2015-09

Model-based systems engineering in the execution of search and rescue operations

Hunt, Spencer S.

Monterey, California: Naval Postgraduate School

http://hdl.handle.net/10945/47277

NAVAL POSTGRADUATE SCHOOL

MONTEREY, CALIFORNIA

THESIS

MODEL-BASED SYSTEMS ENGINEERING IN THE EXECUTION OF SEARCH AND RESCUE OPERATIONS

by

Spencer S. Hunt

September 2015

Thesis Advisor:	Kristin Giammarco
Second Reader:	Bonnie Young

Approved for public release; distribution is unlimited

THIS PAGE INTENTIONALLY LEFT BLANK

REPORT DOCUMENTATION PAGE			*Form Approved OMB No. 0704–0188*
Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington, DC 20503.			
1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** September 2015	**3. REPORT TYPE AND DATES COVERED** Master's Thesis	
4. TITLE AND SUBTITLE MODEL-BASED SYSTEMS ENGINEERING IN THE EXECUTION OF SEARCH AND RESCUE OPERATIONS			**5. FUNDING NUMBERS**
6. AUTHOR(S) Hunt, Spencer S.			
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Postgraduate School Monterey, CA 93943-5000			**8. PERFORMING ORGANIZATION REPORT NUMBER**
9. SPONSORING /MONITORING AGENCY NAME(S) AND ADDRESS(ES) N/A			**10. SPONSORING/MONITORING AGENCY REPORT NUMBER**
11. SUPPLEMENTARY NOTES The views expressed in this thesis are those of the author and do not reflect the official policy or position of the Department of Defense or the U.S. government. IRB Protocol number ____N/A____.			
12a. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited			**12b. DISTRIBUTION CODE**

13. ABSTRACT (maximum 200 words)

Complex systems engineering problems require robust modeling early in the design process in order to analyze crucial design requirements and interactions. This thesis emphasizes the need for such modeling through multiple model-based systems engineering techniques as they apply to the execution of search and rescue. Through the development of a design reference mission, this thesis illustrates how a search and rescue architecture can undergo multiple levels of model-based analysis in order to ascertain critical system behaviors. This way, design aspects can be assessed and modified before incurring the costs associated with incorrect implementation. Furthermore, the study seeks to identify which particular modeling techniques are most conducive to the search and rescue domain. Then, other modelers can build upon the work presented here in order to assess any architecture aspect from different operating procedures to emerging technologies. Ultimately, this study demonstrates that complex systems require multiple iterations across different models. Since each technique has strengths and limitations, it is not feasible to encapsulate every interaction without constructing multiple models. Systems engineering is constantly iterating and seeking to improve designs. Model-based systems engineering can help designers improve not only a search and rescue architecture but also any system today and in the future.

14. SUBJECT TERMS model-based systems engineering, search and rescue, modeling, architecture, sequence diagram, hierarchy diagram, IDEF0, spider diagram, activity model, Monterey Phoenix			**15. NUMBER OF PAGES** 127
			16. PRICE CODE
17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UU

NSN 7540–01-280-5500

Standard Form 298 (Rev. 2–89)
Prescribed by ANSI Std. 239–18

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

MODEL-BASED SYSTEMS ENGINEERING IN THE EXECUTION OF SEARCH AND RESCUE OPERATIONS

Spencer S. Hunt
Lieutenant, United States Navy
B.S., United States Naval Academy, 2006

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN SYSTEMS ENGINEERING MANAGEMENT

from the

NAVAL POSTGRADUATE SCHOOL
September 2015

Author: Spencer S. Hunt

Approved by: Kristin Giammarco, Ph.D.
Thesis Advisor

Bonnie Young
Second Reader

Ronald Giachetti, Ph.D.
Chair, Department of Systems Engineering

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

Complex systems engineering problems require robust modeling early in the design process in order to analyze crucial design requirements and interactions. This thesis emphasizes the need for such modeling through multiple model-based systems engineering techniques as they apply to the execution of search and rescue. Through the development of a design reference mission, this thesis illustrates how a search and rescue architecture can undergo multiple levels of model-based analysis in order to ascertain critical system behaviors. This way, design aspects can be assessed and modified before incurring the costs associated with incorrect implementation. Furthermore, the study seeks to identify which particular modeling techniques are most conducive to the search and rescue domain. Then, other modelers can build upon the work presented here in order to assess any architecture aspect from different operating procedures to emerging technologies. Ultimately, this study demonstrates that complex systems require multiple iterations across different models. Since each technique has strengths and limitations, it is not feasible to encapsulate every interaction without constructing multiple models. Systems engineering is constantly iterating and seeking to improve designs. Model-based systems engineering can help designers improve not only a search and rescue architecture but also any system today and in the future.

THIS PAGE INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF ACRONYMS AND ABBREVIATIONS

ACO	aircraft coordinator
AOR	area of responsibility
C2	command and control
DOD	Department of Defense
DRM	design reference mission
EPIRB	emergency position indicating radio beacon
IAMSAR	International Aeronautical & Maritime Search and Rescue Manual
LKL	last known location
MP	Monterey Phoenix
MBSE	model-based systems engineering
NSARC	National Search and Rescue Committee
NSP	National Search and Rescue Plan (of the U.S.)
NSS	National Search and Rescue Supplement (of the U.S.)
OPSIT	operational situation
OSC	on-scene commander
PID	person(s) in distress
RSC	rescue sub-center
RTB	return to base
SA	SAR assets
SAR	search and rescue
SAROPS	Search and Rescue Optimal Planning System
SITREP	situation report
SMC	search and rescue mission coordinator
SRR	search and rescue region
SRU	search and rescue unit
SysML	Systems Modeling Language
UML	Unified Modeling Language
VFR	visual flight rules

THIS PAGE INTENTIONALLY LEFT BLANK

EXECUTIVE SUMMARY

As technology evolves and advances, systems engineering problems become more complex. In order to understand important system interactions and behaviors, designers need appropriate tools to simplify and assess what designs are doing and what improvements must be made for future implementation. Discovering these insights early is critical to avoiding cost and schedule pitfalls in later stages of system development. Nowhere are these complexities more evident than in a search and rescue (SAR) architecture. Since a large number of system interactions must occur precisely in order to achieve mission success, a SAR architecture presents a uniquely complicated case study in utilizing model-based systems engineering techniques to understand the assets and the mission space.

As such, the purpose of this thesis is to apply various model-based systems engineering techniques to a SAR architecture in order to discover what techniques are most conducive to modeling in the SAR domain, while simultaneously gaining insight on capturing complex system behaviors for any real-world problem. Additionally, this thesis seeks to provide a baseline of modeling work that can be utilized by anyone wishing to explore an aspect of the SAR architecture in order to evaluate new procedures, emerging technologies, asset compositions, or any other area of performance that could improve the SAR system's ability to save lives. As a byproduct, this thesis evaluates the applicability of the various models presented as a means of ascertaining areas where techniques could be improved or supplemented by another technique.

The thesis objectives were accomplished through three distinct phases. The first phase involved a background study of the current organization of SAR infrastructure in the United States. The background information was necessary for understanding how SAR is currently conducted, and it provided a framework for the assets and interactions that would be included in the mission and the modeling.

The second phase was to develop a design reference mission (DRM) to demonstrate the various modeling techniques. This included development of a generic mission narrative that stepped the architecture through a sequential series of interactions along with a set of general rules applicable to all of the models in order to maintain some simplicity through assumed interactions throughout. The DRM also contains four real-world operational situations (OPSITs) for the reader to consider as, inevitably, all models need realistic data inputs in order to assess performance.

With the DRM established, the final phase was constructing the models. Each model was developed using the mission narrative, but insights from other models were utilized as the narrative was translated into each construct. As the process evolved, iteration was natural due to increased understanding of the modeled interactions and behaviors. Thus, models were revisited numerous times for refinement regardless of the order in which they were originally constructed.

Complex systems require a cumulative modeling effort using multiple techniques in order to capture all of the intricacies of system behavior. This was true of the modeling effort in this thesis for the SAR architecture because no one model encapsulated every interaction necessary to evaluate the system. It is not feasible to model everything with one technique because the output of such an effort would be incomplete. Even for this thesis's generic mission narrative, some of the modeling techniques required multiple traces and decompositions just to present one facet of a complicated architecture. Attempting to illustrate all facets at one time would be untenable. This is why so many different modeling techniques exist because each one brings a unique set of strengths and insights despite its individual limitations. Thus, the decision to use a particular modeling technique is dependent on the objectives of the modeler, whether the system is a SAR architecture or any other complex system.

Translating the mission narrative from technique to technique provided a great deal of insight on how to model certain aspects of the architecture that were not apparent at the onset. This is a major benefit of employing multiple modeling techniques because each technique has the distinct potential to illuminate otherwise unpredicted architectural deficiencies. Unpredicted behaviors, whether or not desirable, are best encountered early on so that if the outcomes are undesirable, they can be dealt with immediately. This is the purpose of modeling—to solve real-world problems.

The various SAR architecture models were designed to help future modelers who wish to evaluate their own concepts. The models in this thesis pertain to SAR, but that does not mean that the concepts presented must limit a modeler to that system. If a complex SAR architecture can be understood and analyzed using model-based systems engineering, there is no limit to what can be accomplished by applying these techniques in other areas. It is a worthwhile endeavor, crucial to understanding complex relationships in system design, so that time and money can be saved on incorrect implementation. The goal of all technological progress is to enhance performance and model-based systems engineering is a crucial piece of the design process in reaching such goals.

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGMENTS

First, I would like to thank my wife, Katie, for putting up with me for the last 24 months of this graduate school adventure. I spent many nights making a mess in the office, which often spilled out to other areas of the house, and her patience was crucial as my time staring at the computer screen increased exponentially while working on this thesis. I could not have done any of this without your love and support. I love you and promise not to tackle further education anytime soon.

I would also like to thank my advisor, Kristin Giammarco, for all the help throughout this process. In many ways, I feel like this thesis started a year ago when I first saw her posting on the portal and decided I wanted to be a part of the model-based work she was doing with SAR. Throughout the process, she helped me to understand the direction and scoping I needed to take with my own work, and she has given me crucial insight and guidance through some of the difficult modeling and refinement. Her critiques were spot-on and always timely, meaning that I did not spin my wheels aimlessly for long before making tangible progress. This effort would not have been possible without her.

I also would like to thank Bonnie Young, who did a great job as a second reader, providing valuable insight on some final refinements to the document. Michele D'Ambrosio was essential in getting the thesis through the processing office, and Barbara Berlitz and Aileen Houston made sure I stayed out of trouble with any sloppy citations.

Last, but certainly not least, I would like to thank my mom, Lynne, for taking 10 hours of her weekend to lend her editor's eye to my thesis. Ever since you taught me how to write 18 years ago, I have relied on your expertise for important papers. I love you and promise to you, as well, that I will not do more school anytime soon.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

Search and rescue (SAR) is a global activity with implications spanning both civilian and military entities. The most important factor in any SAR is time. The longer an individual is exposed to the elements—water, hazardous wildlife, extreme weather, extreme temperatures, lack of food and water—the more the chance of survival decreases. Current SAR doctrine is designed around efficient execution for this reason, but with so many elements performing various roles and having different responsibilities, it is difficult to ascertain what mix of units in a SAR architecture works best for any given scenario.

Further complicating SAR missions is the challenge of obtaining successful command and control, effective communication, and management of the overall SAR problem. Civilian and military SAR publications provide methods for determining the most suitable search patterns based on scenario, environmental conditions, and the characteristics of the search object. Simulation software, such as the U.S. Coast Guard's SAROPS, calculates Monte Carlo probability distributions for search object location and can even recommend a basic search plan along with an estimated probability of success based on assets available (Metron Incorporated 2007). Unfortunately, none of these methods can fully encapsulate every function that must execute in a SAR mission. As good as these methods are for individual crews executing searches, they do not investigate components such as communication, command and control, or decision-making. Nor do they assess an optimal configuration of assets to utilize for the mission. Fully evaluating a SAR evolution on a large scale requires robust modeling so the best possible SAR architecture can be realized. Obviously, every SAR situation is different, so identifying the optimal modeling methodology is imperative. That way, any changes in the scenario can be adjusted in the model to ascertain a predictive outcome based upon the SAR architecture in play on the mission.

To that end, the purpose of this study is to develop a baseline Design Reference Mission (DRM) that will be analyzed through various model-based systems engineering techniques and languages in order to determine which technique, language, or combination thereof best mirrors the realities and complexities of SAR. This will be accomplished through an analysis of how SAR is organized and already conducted by various entities, specifically those in the United States. This information will aid in developing the DRM and interpreting the models. Once the DRM is established, it will be modeled using various systems engineering techniques and languages to determine what works best for the SAR mission. From there, mission planners can take the baseline models and adjust them as necessary to evaluate a collection of variables from real-time scenario disruptions and weather, to asset availability and even emerging technologies. Written as a capability needs statement, civilian and defense agencies need an accurate and effective means of modeling SAR operational architectures across multiple scenarios in order to assess current and future capabilities so that persons in distress can receive aid in the shortest time possible.

It is important to note that this study does not assume any specific weaknesses in current SAR doctrine. As such, the study does not make any conclusions about improvements or changes in how SAR is conducted. Instead, the SAR mission architecture serves as a complex case study that provides the backdrop for developing and understanding how to use model-based systems engineering to analyze systems. Ultimately, the goal is for all designers to provide the most effective system architectures. Establishing a methodology that comprehensively analyzes the optimality of an entire architecture is crucial to enhancing this capability today and in the future.

II. RESEARCH OBJECTIVES AND METHODOLOGY

In order to keep the study organized, it is important to establish an outline of objectives along with a planned methodology that will accomplish the objectives. This way, plans and processes can be checked against the objectives to ensure that all work tracks toward the overarching goals. To that end, this chapter presents a set of research objectives and corresponding methodology for the capability need statement: civilian and defense agencies need an accurate and effective means of modeling SAR operational architectures across multiple scenarios in order to assess current and future capabilities so that persons in distress can receive aid in the shortest time possible.

A. RESEARCH OBJECTIVES

The first research objective is inherent to the capability need statement, to find the modeling technique, language, or combination thereof that best mirrors the realities and captures the complexities of the SAR mission. There are a number of techniques and languages, and the assumption is that there is an optimal way to model the mission, along with the assets involved and their interactions, in order to evaluate the effectiveness of the entire SAR architecture.

The second research objective is to create a base model in order to test each technique and language for the evaluation process. The base model will not change throughout the modeling process, so it will provide a level of standardization and equality as each technique is evaluated. The hope is that the iterations of the base model for each technique and language will provide a starting point for anyone who wishes to pursue his or her own evaluation of a different concept. In this way, new procedures and emerging technologies can be evaluated beyond the scope of this study in various future implementations.

The third research objective involves discovering what improvements can be made on the various modeling techniques and languages. While many of the techniques and languages are firmly established and have been used for years, a few are still under development. Finding ways to improve the developing methods, or perhaps even expand the incumbent ones, will benefit the wide array of individuals who use them. Even something as simple as a new way to use a particular technique or language can be extremely beneficial.

The fourth research objective is to gain insight and understanding in model-based systems engineering as it pertains to a real-world mission, scenario, or architecture. While this objective may be obvious, it is helpful to state it here because all three aforementioned research objectives are directly related to it. The purpose of any model is to understand what is being modeled and while no models are perfect, they are one of the best tools of evaluation available without having to perform operational testing. This speaks to the applicability of model-based systems engineering not just for engineers or designers, but also for mission planners and operators. (See Table 1 for a condensed summary of the research objectives.)

Table 1. Research Objectives Condensed Summary List

RESEARCH OBJECTIVES
Find the modeling technique or language that best models the complexities and realities of the SAR mission
Create a flexible and robust base model for future implementation
Discover improvements to be made in current modeling techniques and languages
Gain insight and understanding in model-based systems engineering pertaining to real-world problems

B. METHODOLOGY

The research objectives will be achieved via three main phases: background study, design reference mission development, and modeling. Each phase is described in detail below.

1. Background Study

Understanding the SAR mission is imperative before any useful modeling can be undertaken, so a background study is necessary. Over the years that nations have coordinated with one another to accomplish worldwide SAR, a number of charters and publications have been enacted to establish participants, organizational hierarchies, cooperation plans, responsibilities, and even specific procedures and communications. In the publication realm, many “big-picture” charters and conventions apply across multiple nations and each of those nations usually has its own supplemental documents delineating responsibilities and actors within each sovereign’s area of operation. Since the international-level documents are intended to be universally applicable, a high degree of standardization exists among participating nations despite each nation having its own supplemental documents tailored to individual infrastructures. This is beneficial because examining one nation’s publications provides the necessary knowledge to aid in understanding the SAR mission.

Therefore, in the background study, specific attention is given to how SAR is organized and what the command structure looks like. Additional important pieces include an investigation of lower-level documents that establish participating agency responsibilities, operational procedures, assets in the SAR system, and their various capabilities in the mission space. Ultimately, the goal of the background study is to solidify an understanding of what happens the moment a distress call is received by the SAR system and the interactions by various actors in the system that occur in order to accomplish the mission from start to finish. This is an important point because the modeling focuses on what happens when the SAR system is in action. Without a clear understanding of how the system works, it will be difficult to create an accurate and realistic model, and even harder to gain any insight into the mission or modeling techniques.

2. Design Reference Mission Development

Once the background study has provided context and understanding for the SAR mission and its various entities, it will be necessary to develop a design reference mission (DRM). The DRM is the base mission model that will be used throughout the various techniques and languages in keeping with the second research objective. The development occurs in a few distinct stages, the first of which is outlining some SAR operational situations, or OPSITs. The OPSITs themselves are analogous to individual mission scenarios that provide real-world context. They each tell a story about what kind of potential SAR mission could be expected and they outline specific details regarding variables such as location, environmental conditions, survivor conditions, assets available, and command structure. When the OPSIT is detailed, there is a much better chance for a realistic model that can provide some tangible insights. While multiple OPSITs are not essential, examining a variety of them enhances the applicability of the modeling across multiple scenarios and ensures that the outputs make sense and that a reasonable number of different assumptions have been considered.

The next step in the DRM development involves the establishment of a mission narrative. The narrative is a sequence of "if-then" statements that guide the interactions of the SAR system in the mission. The statements act as decision nodes for each entity involved and based on how certain aspects of the scenario progress, they will lead the actors down different paths toward concluding or continuing the mission. The narrative is intended to be as general and solution-neutral as possible because the sequences from the narrative are the foundation for the base model that will be used for the various modeling techniques. The specificity comes later when looking at an output model and then applying the OPSIT to what was generated. In other words, the narrative is a set of steps that will apply in every SAR mission regardless of individual OPSIT because it is sufficiently inclusive, yet general, in order to be broadly applicable. The general narrative sequence is then modeled and once the output is achieved, individual OPSIT information can be applied and tested in order to evaluate how well a particular model performed. Of note, there will be certain aspects of every mission that cannot be modeled because of feasibility or simplicity concerns. These aspects will be addressed using some general rules that will apply across every mission regardless of whether or not they appear as a sequence or decision node in the modeling. Further discussion on OPSITs, mission narratives, and general rules is contained in Chapter IV.

3. Modeling

When it comes to modeling the problem, the first step will be taking the mission narrative and integrating it into a base model sequence diagram. More than simply translating the words of the narrative into a visual form, the base model sequence diagram establishes the entities that will have input and output, and organizes them according to what those inputs and outputs are and in what order they occur. The base model also serves as the ideal mission where everything goes according to plan, and the objective is accomplished from beginning to end with no unusual or dynamic circumstances changing the outcome. This is important because as the base model is analyzed through different modeling techniques, one form of evaluating a particular technique's usefulness and flexibility is determining how well it handles disruptions, abstractions, and uncertainties that may deviate from the ideal case scenario.

Regarding modeling methodology, each different technique or language will necessitate some study and practice prior to attempting to model the SAR problem. As such, a background and purpose description will accompany every separate modeling section prior to presenting the results. This will aid in understanding each technique so that anyone examining the results can interpret the work regardless of expertise level in any particular technique. Next, because one modeling technique can provide insights into other models, the plan is to accomplish all of the modeling somewhat simultaneously. This means that all models are subject to continuous iteration until the entirety of the modeling effort has been completed. That way, any insights gained throughout the process of building other models can be implemented across one or all of the other models under consideration. This is part of the iteration and refinement process, and something that should not be overlooked in examining all of the modeling. When all of the models are complete, it will be easier to assess the different techniques objectively and comprehensively as they pertain to the SAR mission space and their applicability not just to SAR, but to model-based systems engineering as a whole.

THIS PAGE INTENTIONALLY LEFT BLANK

III. SEARCH AND RESCUE ORGANIZATION AND PROCEDURES

Early documentation of SAR dates back to 1655 when three separate searches were initiated by the Dutch to recover men and goods from the ship *De Vergulde Draeck*, which wrecked on a reef near Australia in April of that year. According to logbook accounts, SAR in those days consisted of merely sending more surface assets to the last known location of the vessel in question and trolling the coastline near the reef for any signs of wreckage (Major 1859, 77). Such practices had little success, yet the hope of rescue and recovery warranted further risk to personnel and assets. This mindset has not changed over the years as humans have attempted to perfect the practice of SAR.

Today, the organization and execution of SAR is much more formalized thanks to a number of cooperating nations that have developed guidelines, procedures, and structure for the mission. The idea of international cooperation and similarity of structure and procedure is predicated on the idea that no single entity has sufficient SAR resources to provide adequate services in every situation and as such, a coordinated effort is needed. Thus, in 1979, "noting the great importance attached in several conventions to the rendering of assistance to persons in distress at sea and to the establishment by every coastal State of adequate and effective arrangements for coast watching and for search and rescue services," the International Convention on Maritime Search and Rescue was convened (United Nations [UN] General Assembly 1979, 119). The treaty that resulted from the convention established everything from governing bodies and responsibilities to procedures, and provided the framework to further develop and promulgate specific manuals and procedures (UN General Assembly 1979, 227). One such manual is the *International Aeronautical and Maritime Search and Rescue Manual* (IAMSAR), which is jointly published by the International Maritime Organization and International Civil Aviation Organization. The IAMSAR is a three-volume set that "provides guidelines for a common aviation and maritime approach to organizing and providing SAR services" (International Maritime Organization 2015, para 2). Stemming from the 1979 treaty, the regularly updated and re-published IAMSAR is the international SAR community's primary governing document. In fact, international safety of navigation regulations requires ships to carry an up-to-date copy of Volume III of the IAMSAR because it contains specific procedures to aid ships and aircraft with their responsibilities as SAR assets, as well as their own emergencies (International Civil Aviation Organization and International Maritime Organization 2013, iii). A comprehensive examination of the IAMSAR is beyond the scope of this study. Rather, this study will focus on a single participating nation in order to establish an organizational and procedural baseline for the DRM and subsequent modeling. Since the United States has been involved significantly in the international SAR community for some time, United States manuals and procedures will serve as this baseline. Examining the United States' National Search and Rescue Plan, The National Search and Rescue Committee's supplement to the IAMSAR and various U.S. Department of Defense manuals will provide the background in purpose, structure, capability, and understanding needed not only for the DRM development but for the modeling itself.

A. NATIONAL SEARCH AND RESCUE PLAN OF THE UNITED STATES

The purpose of the United States' National Search and Rescue Plan (NSP) is to establish the "effective use of all available resources in all types of civil SAR missions to enable the United States to satisfy its humanitarian and national and international legal obligations" (United States National Search and Rescue Committee [NSARC] 2007, 1). This means that the NSP does not promulgate specific procedures on how to conduct a rescue, but rather outlines an over-arching design for how the United States fulfills its role in both domestic and international SAR. In all cases, the national plan does not supersede or conflict with any responsibilities delineated in international governing documents such as the IAMSAR or the 1979 convention but rather provides further instruction and clarification for U.S. entities as they coordinate with other SAR agencies at home and abroad. Some of the specific objectives of the NSP include pursuing efforts to improve asset cooperation, providing national guidance for development of civil SAR-related systems, describing SAR participants, and researching and developing procedures, technologies and coordination in order to improve overall SAR services (NSARC 2007, 3).

1. Participants and Responsibilities

An important part of the NSP is the establishment of structure and organizational entities that have authority to execute and promulgate policy and procedure. The National SAR Committee (NSARC) is the main component of the organizational hierarchy and is responsible for the provisions of the national plan along with coordinating and providing guidance for its implementation (NSARC 2007, 4). NSARC has a number of member agencies that the NSP calls participants and they each fulfill a specific role within the overall organization of NSARC. For instance, the Department of Homeland Security is a participant whose responsibility is to respond to hazards and distress situations affecting the United States and its people. This is accomplished by the U.S. Coast Guard and the Federal Emergency Management Agency (NSARC 2007, 4). The Department of Transportation is also a participant, carrying out its responsibilities in transportation safety through the Federal Aviation Administration and the Maritime Administration. Both branches of the Department of Transportation establish and enforce safety regulations. While the Federal Aviation Administration does not maintain a fleet of ready aircraft for government use like the Maritime Administration, it does operate air traffic control, navigation and flight service facilities that are available around the clock to contribute to SAR operations (NSARC 2007, 4). Other participants include the National Oceanic and Atmospheric Administration, the Federal Communications Commission and its long-range direction finder network, the National Aeronautics and Space Administration, and the National Park Service through the Department of the Interior, which provides services over the interior lands and waters of the United States (NSARC 2007, 5). The final major participant is the Department of Defense (DOD). Its participation is unique in that its resources are "used for civil SAR needs to the fullest extent practical but on a non-interference basis with primary military duties" (NSARC 2007, 11).

Depending on what situation the military is involved in, the NSP specifically delineates what SAR services are covered by the national plan and which services would be delegated to another entity governed by an outside charter (NSARC 2007, 11). For instance, the NSP covers SAR services for maritime, aeronautical, land, urban, disaster and distress situations, medical transportation, saving property in conjunction with saving lives, mass rescue operations, and SAR services for incidents of national significance (NSARC 2007, 11). The national plan does not cover air ambulance services, rescues from space, military operations such as combat SAR or removing personnel from harm, salvage, assistance with civil disturbances, or terrorism response (NSARC 2007, 11). Many of the services not covered by the national plan are handled by the DOD in full or in part, which is why their services in civil SAR fall under the non-interference clause. Furthermore, because the military owns and maintains its own resources related to SAR operations, the military is responsible for executing its own SAR missions and even has its own supplements to the SAR publications promulgated in the civil domain. A thorough discussion of services not covered by the NSP is beyond the scope of this study. However, it is worth noting that due to certain procedural similarities, those services also could be modeled using adapted versions of the systems engineering techniques presented later.

2. Search and Rescue Regions

In addition to establishing participants and responsibilities, the national plan draws out different contiguous, non-overlapping SAR regions (SRR). The reason for this is “to ensure provision of adequate land-based communications infrastructure, efficient distress alert routing, and proper operational coordination to effectively support civil SAR services” (NSARC 2007, 5). This is understandable because an organizational network as large as global SAR needs boundaries so that participating entities know where their areas of responsibility lie and what kinds of resources are needed in order to provide adequate services in their respective SRR. Beyond entities operating from the United States, the SRRs also effect an understanding between all nations concerning where they accept primary responsibility for providing civil SAR service. Thus, the United States’ SRRs are harmonized with those of other nations so that responsibilities do not overlap. In all cases, the existence of SRR boundaries do not exist to restrict or delay prompt action to render aid if the opportunity arises for assets not directly associated with a particular SRR.

Each SRR is managed by a SAR Coordinator who has "overall responsibility for establishing Rescue Coordination Centers (RCCs) and providing SAR services" (NSARC 2007, 6). The RCCs are the main component of command and control within an SRR and the national plan stipulates that the RCCs must be staffed with trained personnel 24 hours a day. There is only one RCC associated with each recognized SRR but that does not preclude SAR Coordinators from establishing rescue sub-centers (RSCs) to share responsibility and workload and to complement the overall effort in the SRR. The United States currently has three SAR Coordinators—the Air Force, U.S. Pacific Command and the Coast Guard. The Air Force is the SAR Coordinator for the United States aeronautical SRR corresponding to the continental United States (not including Alaska). Pacific Command covers the aeronautical SRR for Alaska, and the Coast Guard covers all other United States aeronautical and maritime SRRs, including Hawaii and all waters over which the country has jurisdiction (NSARC 2007, 7). Because the SRRs and RCCs are integrated into the global SAR system, they must comply with established international standards from relevant conventions and the IAMSAR. This is how standardization occurs with participating nations across the globe. Furthermore, it is the reason that modeling efforts predicated on United States entities and procedures can be applied internationally.

In summary, the United States National SAR Plan is an important document that establishes an organizational framework of participating agencies that have specific responsibilities in defined geographical areas. Participating agencies operate their assets underneath SAR Coordinators that are responsible for individual and bounded search and rescue regions that receive command and control direction from the main rescue control center and its subordinate rescue sub-centers. This structure is in compliance with recognized international governing directives so that the United States is fully integrated into the global SAR system.

B. UNITED STATES NATIONAL SEARCH AND RESCUE SUPPLEMENT

The purpose of the National Search and Rescue Supplement (NSS) is to provide guidance and clarification on the implementation of the NSP and the IAMSAR insofar as they pertain to United States operational SAR assets (NSARC 2000, 1–2). It is published by NSARC and contains general policies and procedures that are not specific to a single federal agency. This is so the NSS can remain procedurally general enough to be used across all United States SAR participants, leaving room for individual entities to expand upon policies and procedures to fit their individual areas of expertise. While the NSP has its focus primarily on organizational hierarchy, roles and responsibilities, the NSS focuses on what each entity brings to the mission space and how the different agencies interoperate procedurally in order to execute from initial notification all the way through post-mission debriefing. Thus, the NSP is more of a "big-picture" document laying the foundation and objectives for United States SAR. The NSS then takes that foundation and establishes policy and procedures so that United States entities are accomplishing the NSP.

Observing the mission in action is the focus of the modeling; therefore, this section will focus primarily on the operational aspects of the NSS. Understanding how the mission is executed and what part is played by each asset or entity will provide key insights into how each will interact within the model.

1. Stages of SAR Assistance

The NSS states that SAR services usually result when the SAR system is notified "of a potential or actual distress situation that may involve the need for assistance" (NSARC 2000, 1–2). The NSS refers to these situations as SAR incidents. Any SAR incident can be designated as a SAR case if the situation warrants, but only if the situation evolves to where SAR assets are deployed can a SAR case be designated as a SAR mission. The IAMSAR and NSS both break down the response of a SAR incident into a sequence of five typical stages that define the kind of assistance provided at any given time as the incident unfolds. Each unique incident may or may not include every stage and the stages themselves can overlap depending on the situation. The five stages along with their definitions are outlined in the table below.

Table 2. SAR Five Stages of SAR Response (from NSARC 2000, 1–2)

Awareness:	SAR system becomes aware of an actual or potential incident.
Initial Action:	Preliminary action taken to alert SAR facilities and obtain amplifying information. This stage may include evaluation and classification of the information, alerting of SAR facilities, preliminary communication checks (PRECOM), extended communication checks (EXCOM), and in urgent cases, immediate action from other stages.
Planning:	Effective plan of operation is developed, including plans for search, rescue, and final delivery.
Operations:	SAR facilities proceed to the scene, conduct searches, rescue survivors, assist distressed craft, provide emergency care for survivors, and deliver survivors to a suitable facility.
Conclusion:	SAR facilities return to their regular location, are debriefed, refueled, replenished, provided with a fresh crew, and prepare for another mission; documentation of the SAR case is completed.

The definitions presented above illustrate how the different stages in any given mission can overlap. For instance, if an RCC is notified of a SAR over the water that will require a long transit for any airborne rescue units, the planning phase can occur simultaneously with the operations phase as individuals on the ground can work the search plan while rescue crews proceed to the scene. This is one of many instances where the stages could overlap. It is important to remember that this kind of flexibility is crucial to the mission because of the dynamic environments where SAR missions occur. This is why the stages are not rigid sequential requirements. It will be useful to note this distinction in the models so that they can be as realistic as possible. The NSS has provided a basic framework for the stages of a SAR, so at this point, this study will examine additional specifics regarding the participants in the five stages and how such participants function as actors on the mission.

2. SAR Mission Coordinator and Command and Control

As previously discussed under the NSP, the United States has several SAR regions called SRRs that are each managed by a SAR Coordinator who has the responsibility to conduct civil SAR services in the respective SRR. What the NSP does not address is the fact that the SAR Coordinator normally is not personally involved in any actual coordination or provision of SAR services because the Coordinator's primary function is as an executive-level leader and manager (NSARC 2000, 1–4). Thus, the duties of running the mission will rest on the Search and Rescue Mission Coordinator (SMC) who wields the full operational authority of the SAR Coordinator on any given mission. The SMC will usually reside in the RCC and as such, the NSS stipulates that any SAR mission that involves an RCC should have a designated SMC conducting the operation. The NSS states "SAR operations are normally coordinated at the lowest practical level within the SAR system," (NSARC 2000, 1–3). This is understandable, because not every incident needs the full capacity of an RCC on the case. That is one of the reasons for rescue sub-centers, alert posts, and various staging areas and how each can receive delegation of authority from the SAR Coordinator. To a typical asset out on scene, the difference between controlling agencies is mostly transparent. However, if one is to understand what takes place behind the scenes at the control centers, it is necessary to know who is acting in what capacity. For the purposes of simplification and applicability in the model, all of the workings of the command structure will be synthesized into the singular identity of Command and Control or C2. That way, whether the mission is conducted from a remote Alaskan rescue sub-center or from the parent RCC, C2 will remain the consistent agent of overall command in each situational variation.

3. On-Scene Coordinator and Aircraft Coordinator

A second operational actor in the NSS that is not covered in the NSP is the On-Scene Coordinator (OSC). The OSC is designated by the SMC to manage the SAR operation at the scene, usually when multiple assets are involved in the SAR. The OSC should be the best-qualified person or unit available at the scene, meaning that it may not always be an individual or asset that has any special SAR training. Specific considerations for choosing an OSC should include their SAR training, communications capabilities and length of time they can stay in the search area (NSARC 2000, 1–7). An OSC need not be an aircraft or unit that is even capable of making the rescue because ultimately, that is not the point of having an OSC. The point of the OSC is mainly to improve on-scene coordination because it can be difficult to accomplish from assets that are nowhere near the search area. The NSS also discusses the use of aircraft coordinators (ACO) on scene for the purposes of flight safety. An OSC can fulfill the duties of an ACO but it may make sense to split the duties if there are no communication links between the OSC and participating aircraft, for instance. It is also another way to share tasks if there are so many airborne assets that the OSC becomes overwhelmed. Of note, the OSC and ACO, being designated by the SMC, have the full operational authority of the SMC on scene and are responsible for the entirety of the coordination effort. Some specific duties of OSCs and ACOs include establishing and maintaining communications with all SAR assets on station, establishing common altimeter settings for on-scene aircraft, providing initial briefing and search instructions for arriving assets, receiving and evaluating sighting reports, and submitting situation reports (SITREPs) to the SMC (NSARC 2000, 1–8). Of note, OSC can also refer to an on-scene commander, the difference being nominal only. Having a commander instead of a coordinator usually occurs in SAR where the military is involved, as their publications refer to the OSC specifically as an on-scene commander. The duties and responsibilities of the OSC are identical regardless.

4. SAR System OV-1

Up to this point, there has been a lot of discussion about different assets in the mission space from the SAR Coordinators to the SAR Mission Coordinators, the Rescue Control Centers to the Rescue Sub-Centers and of course the On-Scene Commanders and the various SAR units. Because there are so many actors and acronyms, it is easy to get confused and lose sight of how the SAR system actually works. To alleviate the confusion, Figure 1 is an OV-1 high-level operational concept graphic that presents a visual depiction of how all of the aforementioned actors generally interact within the SAR system.

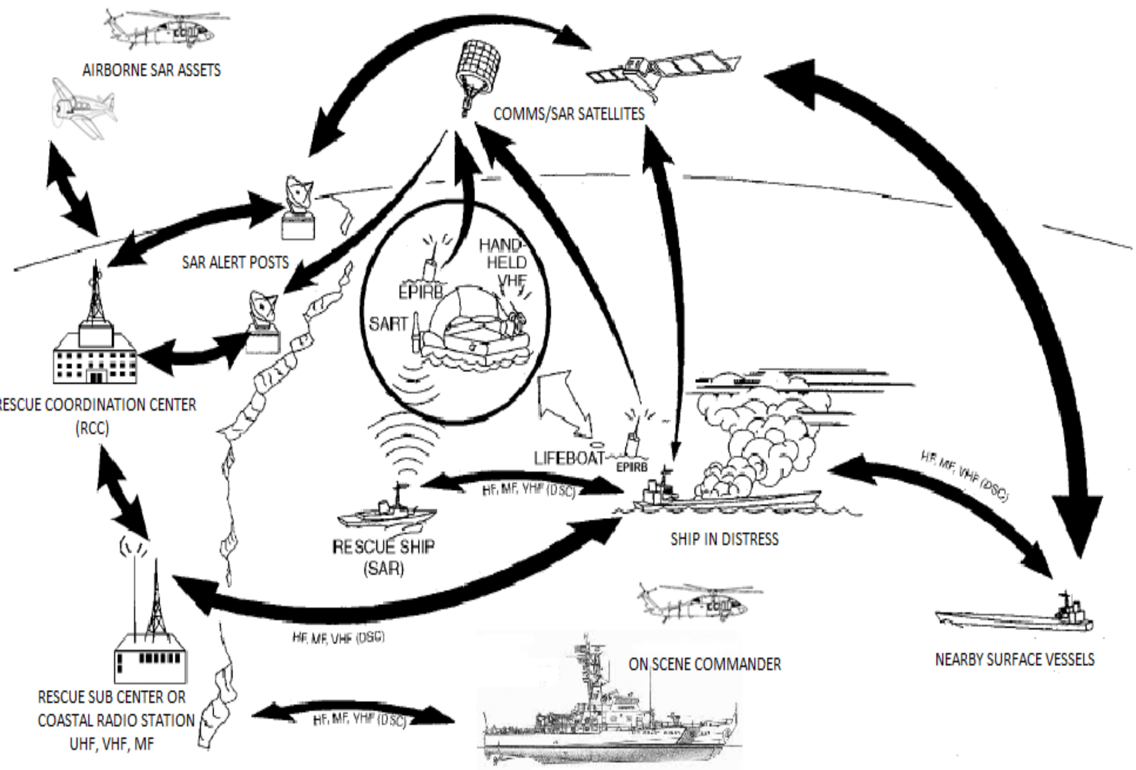

Figure 1. SAR OV-1 (after Johnson and Lee 2011)

This particular OV-1 presents a maritime SAR situation, but it could easily be adapted for a different mission over land. In this scenario, a ship in distress or a person in distress sends out a signal using whatever equipment available. It could be a radio call, a transmission from an electronic beacon or even a visual signal to a nearby surface vessel or aircraft. This distress signal alerts the SAR system to a potential SAR mission and that signal can be received and relayed by any nearby vessel or coastal station, or it can be pushed to various land-based stations via communications satellites. Since the RCC is responsible for a large area, it will not necessarily be physically located anywhere near the mission scene and that is why there are SAR alert posts and a rescue sub-centers positioned along the coast. After the initial notification, the system is in the awareness stage of the mission. The next stage is initial action and that is where the controlling entity that receives the distress call begins to gather information and scramble rescue units as applicable. Whoever takes responsibility for controlling the mission will effectively become command and control, and it will be up to that entity to execute the planning, operations and conclusion stage of the mission. Also depicted are potential OSC units coordinating various assets on scene and providing reports back to C2. Thus, the OV-1 provides a visual context for how the SAR system units generally interact with each other. Specifically, it has shown the role of C2, the person or vessel in distress, SAR rescue assets, the OSC and even the physical environment where the mission could take place. Together, these entities are the main actors in the DRM and subsequent modeling. More specifics on how they interact will be presented in the DRM and mission narrative.

THIS PAGE INTENTIONALLY LEFT BLANK

IV. DESIGN REFERENCE MISSION DEVELOPMENT

The purpose of a main reference mission is to define a specific scenario that is adequately bounded by physical and functional parameters, and that contains the appropriate amount of detail so that measures of mission success can be collected and used to assess a system as a whole (Giammarco, Hunt, and Whitcomb 2015). In the case of this study however, the reference mission will be used to evaluate modeling techniques and languages rather than a specific SAR architecture itself. This does not change the methodology of developing the reference mission or the evaluative progression throughout, but simply the end conclusion. Stated another way, the goal is not to find the best SAR architecture but instead to determine which modeling technique or language is most appropriate for representing the complexities of the SAR mission. Therefore, the analysis must naturally look at some kind of theoretical asset architecture from the perspective of an integral unit. This is necessary so that a mission narrative resulting from the reference mission can have focus and boundaries. Additionally, a core unit that stays the same throughout a set of scenarios provides a consistency that ties not only the various mission situations together, but the modeling as a whole. This idea is further expounded upon below in the description of the DRM operational situations (OPSITs).

The complexity and variation of circumstances in SAR missions makes it impossible to encapsulate every situation in one reference mission. In fact, this would be undesirable for the purpose of the study because an overly complex DRM can lead to an intractable model that is difficult to understand. For the sake of completeness, a reference mission can have several variants that fall within the analysis scope of the DRM so that the DRM ends up containing multiple operational scenarios (Giammarco, Hunt, and Whitcomb 2015). This concept reflects the goal of this study because having a few different operational scenarios within the DRM provides a representative cross-section of missions to test against the various modeling techniques. To begin, consider the following capability need statement: Civilian and defense agencies need a cost-effective means to search large areas of ocean and over-land terrain in various environmental conditions in order to locate wreckage and survivors and provide aid in the shortest time possible. This statement is a slight variation on the one presented in the introduction. The key difference is that the need is scoped specifically to help with the details of the reference mission. Whereas the overarching need is to develop accurate modeling, the specific need here is to develop the DRM to test against the models based upon the mission of searching large areas of ocean and land and providing assistance to those in distress. Since SAR operations reach across civilian and military entities, OPSITs can be detailed in the DRM from both perspectives. In this way, the modeling can include assets, architecture and procedures from the different entities in order to provide a more complete picture on the accuracy of the modeling. As such, two different initiating events—a man overboard and a downed aircraft—will be the basis for the OPSITs considered for the modeling. As previously stated, the modeling analysis will consider the perspective of an integral unit that is available in each OPSIT. Since helicopters are a key component to many SAR missions, they will be the integral unit of the DRM. To provide even more specificity, the analysis will consider only one helicopter in each OPSIT operating in the capacity of on-scene commander (OSC) in order to evaluate interactions with other entities in the scenario such as command and control, other SAR assets, the environment and the persons in distress. Therefore, what follows below are four OPSITs derived from the previously mentioned two initiating events that form the basis of the DRM, which is named “Conduct Wide-Range Search for Wreckage and Survivors.” After the OPSITs is a generic and solution-neutral mission narrative that will be adapted to each OPSIT in order to transition them into the various modeling techniques for analysis.

A. MAN OVERBOARD OPERATIONAL SITUATION

Whether it is a crab-fishing vessel in the Bering Sea or a nuclear-powered aircraft carrier in the middle of the southeast Pacific Ocean, man overboard situations are a constant threat. While these situations can manifest themselves in a number of ways, it is important to remember that the capability need statement for the DRM specifically mentions searching large areas of ocean. Thus, it makes no sense to conjure a scenario that involves a precisely known survivor location because that type of SAR is not in line with the need statement. As such, the following scenarios are two man-overboard

OPSITs, one from a military vessel and one from a civilian vessel. The OPSITs are written in narrative form.

1. Navy Man Overboard Operational Situation

Having finished a straits transit between China and Taiwan, the USS George Washington strike group has been steaming overnight in the direction of Yokosuka, Japan to cap off the end of another yearly deployment. The strike group consists of one cruiser, USS *Shiloh* (CG 67), one destroyer, USS *Mustin* (DDG 89) and the aircraft carrier itself, USS *George Washington* (CVN 73). At approximately 0730 local time, a sailor from USS *Mustin* was reported missing at morning muster by a fellow bunkmate. The subsequent man overboard muster on the ship confirmed that the sailor is indeed no longer on the ship. The last time the sailor was seen was at 2230 the previous night just prior to his final rounds to secure the flight deck. The sailor was wearing a standard float coat containing a sea dye marker, a day and night smoke, flares, a signal mirror and inflatable rubber lobes for flotation. The sailor does not have any anti-exposure protection. The average ambient temperature is 90°F (32°C) and the sea surface temperature is 82°F (28°C). Sea state for the last 12 hours was reported at 2 on the Douglas Sea State Scale (1-3 feet of wave height). Current conditions are sunny with a visibility of 10 miles and winds out of the north at 5 knots. From 2230 to 0300, the strike group was steering 090 magnetic and at 0300 made a turn north to 020 magnetic and have been on that course ever since. The strike group's speed has been constant at 20 knots. There have been no hits off of the EPIRB on the sailor's float coat. The assets available are all three ships, which each have a rigid inflatable boat rescue crew, two MH-60S helicopters and one E-2C Hawkeye off of the aircraft carrier. Both helicopters are fully equipped with a rescue swimmer and all the necessary equipment to execute a SAR and Lightning 617, the senior crew of the two, has been tasked as the OSC.

2. Civilian Man Overboard Operational Situation

It is the middle of another king crab season in the Bering Sea. At 2145 local time, a distress call is received from a fishing vessel that has caught on fire in heavy seas due to malfunctioning equipment. At that time, a set of approximate coordinates were transmitted. At 2300, the final distress call stated that all six crewmembers were abandoning the ship due to the out-of-control fire. The final distress call does not contain an updated set of coordinates. All six crewmembers are equipped with anti-exposure protection against hypothermia, personal flotation, flares, smokes and a raft that fits all six. Current conditions are overcast with a ceiling of 1000 feet and a visibility of seven miles. Illumination levels for the night are at 23% and winds are heavy out of the west at 50 knots. Ambient air temperature is -7°F (-22°C), the wind chill temperature is -43°F (-41°C) and sea surface temperature is 38°F (3°C). Sea state is a seven on the Douglas Sea State Scale (20–40 feet of wave height). Available assets include one Coast Guard SH-60F helicopter that is 90 minutes away from the original set of transmitted coordinates along with two other nearby fishing vessels that are 20 and 35 nautical miles away respectively from the original set of transmitted coordinates. The Coast Guard helicopter, Calumet 610, is fully SAR capable and has been tasked as the OSC.

B. DOWNED AIRCRAFT OPERATIONAL SITUATION

Another common SAR initiating event is that of a downed aircraft. Most modern aircraft have onboard equipment that transmits location data to controllers, and in all controlled airspace, radar operators have real-time data on display to show precise aircraft locations in space. If this were not enough, pilots are required to file flight plans with various agencies so that if something were to go wrong, rescuers would have an intended route of flight at a minimum. For all these safeguards however, aircraft still go missing for various reasons to include malfunctioning equipment, bad weather and flights into uncontrolled airspace (i.e., long transits over water or low-level routes through mountainous terrain). For this reason, downed aircraft scenarios are a great fit to the need statement of searching large areas of water and land for wreckage and survivors. Like the man overboard situation, the initiating event of a downed aircraft will be viewed from both a military and civilian scenario to ensure completeness within the DRM. The downed aircraft OPSITs are also presented in narrative form.

1. Navy Downed Aircraft Operational Situation

Two F/A-18E Super Hornets were practicing gun maneuvers during normal carrier cyclic operations. Their practice runs were being conducted on a Mk-58 marine location marker smoke that was dropped off the aircraft carrier's 120 radial for 51 nautical miles. At approximately 1430 local time, there was a mid-air collision of the two aircraft at 9000 feet above ground level. Before ejecting, one pilot made a mayday call but provided no coordinates or position update. There have been no radio transmissions from survival radios and no hits off any emergency location devices. Current conditions are sunny with haze, no cloud layer and eight miles of visibility. Winds aloft are 270 degrees at 19 knots. Ambient air temperature is 79°F (26°C), sea surface temperature is 68°F (20°C) and sea state is a three (three to five feet of wave height) on the Douglas Sea State Scale. Each pilot is equipped with personal flotation and a survival vest that contains sea dye, signal mirrors, day and night smoke, flares and a reflective helmet. Neither pilot has anti-exposure equipment. Available assets are two airborne MH-60S plane guard helicopters, one E-2C Hawkeye and a destroyer that is 20 nautical miles east of the aircraft carrier. Both helicopters are fully SAR capable and the most experienced crew, Lucky 620, is tasked as the OSC.

2. Civilian Downed Aircraft Operational Situation

A private pilot filed a low-level VFR flight plan from a Colorado ski resort back to his home airfield. The departure point was Crested Butte Regional Airport and the destination was Centennial Regional Airport, just outside of the city of Denver. Total flight distance measured approximately 200 miles with a flight time of approximately 90 minutes. The pilot's intended route included several visual checkpoints from the chart. Approximately 2 hours past the filed landing time and after several attempts to contact the aircraft via emergency frequencies, the local flight service station initiated its missing aircraft protocol. The last agency to speak with the aircraft was the tower controller at Crested Butte Regional upon departure at 1200 local time. Current time is 1530, weather is sunny with 10 miles of visibility. Winds are variable at 15 knots gusting to 25 knots. Ambient air temperature is 23°F (-5°C) with an overnight low of 5°F (-15°C). The type of aircraft is a 4-seat Piper Warrior II prop plane and the manifest states three people on board. The status of survival equipment on the aircraft is unknown and there have been no transmissions from any emergency beacons. Search assets on hand include two fully crewed Bell-207 rescue helicopters, two Cessna-152 fixed-wing propeller planes and various ground units located anywhere from 20 to 50 miles away all along the intended route of flight. Due to their capabilities and experience, a seasoned crew from one of the helicopters, call-sign Landslide 07, is tasked as the OSC.

C. DRM MISSION NARRATIVE

Now that the OPSITs are established, the DRM must be decomposed into the individual operational activities or tasks that will constitute executing the mission. This execution involves multiple entity "nodes" that all act simultaneously and in conjunction in order to complete the mission successfully. It is important to keep these actions and entities as solution-neutral as possible since the point of the analysis is not to presuppose a solution throughout but to allow the execution sequences to stand as a baseline for comparing multiple concepts (Giammarco, Hunt, and Whitcomb 2015). The mission narrative must also necessarily be independent of solution if it is to apply across the four OPSITs under consideration. For the purposes of this particular study, the various concepts will be the collection of modeling techniques rather than competing concept architectures. Once again, this does not significantly affect the progression through the process, but simply the conclusion. Whereas a typical mission narrative will include multiple possible paths through mission execution to evaluate concept variants, the narrative in this study will consider a relatively focused set of parameters and "if-then" statements to determine if any modeling technique is superior to the others for modeling the complexities of SAR. Once established, others can take the modeling and use it to analyze their own concepts. To that end, the following is the mission narrative for the DRM "Conduct Wide Range Search for Wreckage and Survivors." It is written with "if-then" statements to show possible progressions through mission execution as the various nodes perform their actions and will form the basis for modeling all four OPSITs. The sequenced steps are numbered to ease in the translation of the narrative into various models. Any events that are recurring or that can occur at any point during the mission are denoted separately from the “if-then” sequence as general rules. This is helpful in this study because it simplifies modeling diagrams since the general rules are always in play and need not be included in every visual model. They are also useful for individuals evaluating different concepts because they allow for multiple iterations to be tested, as instances of the general rules can easily be injected at any point in the mission scenario.

1. Mission Narrative

1. Command and control (C2) either receives a distress signal from a person in distress (PID) or is notified of a missing person or vessel.
2. If the general location information falls outside of the C2’s area of responsibility (AOR), the mission is assigned to the appropriate entity. If the general location is within the C2’s AOR, C2 initiates SAR protocol and passes mission information to available assets.
3. Search and rescue units (SRUs) deploy to the search area as assigned by C2 and the designated on-scene commander (OSC) attempts to contact the PID or the missing person or vessel. If contact is made, the OSC requests a precise location and situation report (SITREP). If no contact is made, the OSC will periodically try again.
4. Upon reaching the search area, datum or last known location (LKL), OSC initiates a search pattern based on the mission situation to include environmental conditions, available assets, crew composition and time on station.
5. OSC conducts the search plan and all assets involved in the search pattern scan the environment for any signs of the PID or vessel. All other SAR assets (SA) report directly to OSC and OSC provides regular SITREPs to C2.
6. If any SA spots an object of interest, that SA maneuvers for a closer inspection. If the object of interest appears to be wreckage, the SA notifies OSC and OSC notifies C2 of the situation. If object of interest appears to be a PID, then the SA notifies OSC and maneuvers to rescue or has OSC coordinate with another SA to make the pickup. If the object of interest is not related to the SAR mission, the SA resumes the search pattern until spotting another object of interest or conditions are reached for a return to base (RTB).

2. General Rules

- Throughout the mission, all assets constantly monitor bingo conditions—the point at which the unit is no longer SAR capable and has just enough fuel remaining to execute a safe and successful RTB—and provide on-station time updates to OSC who communicates with C2. As an SA approaches bingo conditions, they will request a replacement if available and upon its arrival, executes an RTB.

- At any point in the mission, the OSC may receive SITREPs or maneuvering commands from C2.
- At any point in the mission, the OSC may request a SITREP from C2, especially if a significant length of time has elapsed since an update was received.
- At any point in the mission, if the OSC receives information containing the location of the PID or vessel, then it confirms receipt, proceeds to the LKL or tasks an SA to proceed to the LKL and provides a SITREP to C2.
- At any point in the mission, if an SA experiences a condition or system failure rendering it unsafe or ineffective at accomplishing the mission, the SA notifies OSC and executes an RTB. OSC will coordinate with C2 for a replacement SA as applicable.
- If survivor(s) are found, the SA provides the condition of each survivor rescued to OSC who will pass the information to C2 so that medical follow-on treatment can be coordinated.
- In multiple survivor situations where survivors are separated, if a rescued survivor provides updated information on the location of other survivors, the SA notifies OSC, OSC notifies C2 and OSC and adjusts the search plan and pattern as necessary to include the new information.
- In cases where the OSC is the only SA on station, the OSC assumes all mission responsibilities outlined above including making the rescue if able. If unable, OSC remains on station as long as possible for coordination and assistance until an SA arrives that can make a rescue or the OSC is relieved by a more capable platform.

V. DESIGN REFERENCE MISSION MODELING

Now that the design reference mission is established with a clear mission narrative and a set of general rules, the next step is to begin the modeling process. To stay consistent with the methodology presented in Chapter II, the modeling discussion starts with the mission narrative modeled via sequence diagram. From there, the narrative is expanded across various systems engineering models to include an executable simulation model. Throughout the process, the models have taken on a number of iterations as new insights were gained from various techniques and outputs. These insights and iterations will be discussed in the individual model sections of the chapter and will be drawn upon heavily in assessing the merits of the various techniques.

A. SEQUENCE DIAGRAMS

Sequence diagrams have their origins in Unified Modeling Language (UML). Unfortunately for systems engineers, the UML construct tends to be somewhat software-centric so the International Council on Systems Engineering decided in 2001 to customize UML for systems engineering applications (Bell 2004, 1). The result was Systems Modeling Language (SysML), which is comprised of a number of different models—including sequence diagrams—that can be used for systems engineering analysis. When analyzing a system or a process, the sequence diagram is a logical place to start because it is an interaction diagram that shows how processes, components, or entities operate with each other and in what order the interactions occur in order to achieve a desired outcome. This is helpful for visualizing various scenarios in a graphical fashion and for the SAR problem, it shows exactly how the SAR system entities interact sequentially under the mission narrative.

A sequence diagram starts with identifying the assets that have roles in the system. These assets could be various components of a computer system in a car, interacting mechanisms on a piece of construction equipment, or even independent entities in a large system of systems concept architecture like that of the SAR system. A sequence diagram is extremely helpful because no matter how simple or complex the system is, if it can be broken down into different components and their interactions, it can be modeled with a sequence diagram. After identifying the assets, each one is labeled and placed inside a box at the top of the diagram. Vertical lines known as lifelines extend downward from each asset, demonstrating the role of each asset through the sequence. Horizontal arrows known as messages connect the lifelines of the assets and are ordered so that the first message connoting an interaction is at the top of the diagram. Wording that describes the interaction is placed on top of the message arrow. While the wording of the messages is not the focus of the sequence diagram, the messages should be written carefully so that they concisely convey a specific action or event. This way, they will not be confused with other interactions, nor will they clutter the diagram with too many words. Once the assets and lifelines are in place, completing the sequence diagram is simply a matter of stepping through the sequential system interactions and properly ordering and connecting the horizontal messages between the assets.

1. Model

Figure 2 is a sequence diagram for the DRM titled Conduct Wide Range Search for Wreckage and Survivors. As mentioned in the methodology, this sequence diagram serves as the ideal case where no disruptions occur in the mission. This means that each one of the sequential steps occurs in order from the inception of the distress call to the rescue and conclusion of the mission. This ideal case is the best place to begin with the sequence diagram because it establishes the simplest and shortest path to mission success so that any subsequent cases will simply be deviations from the base diagram. Each message that connects the lifelines of the assets inherently has an “if-then” component consistent with the mission narrative and general rules so that at any point in the base diagram, a whole new set of conditions could arise that would lead the mission down a completely different path. When analyzing these different paths, the base model is even more important because it provides an anchor point of comparison for evaluating the effectiveness of the architecture across numerous missions that deviate from the ideal case. This is particularly important for executable models, which will be discussed further on in the chapter.

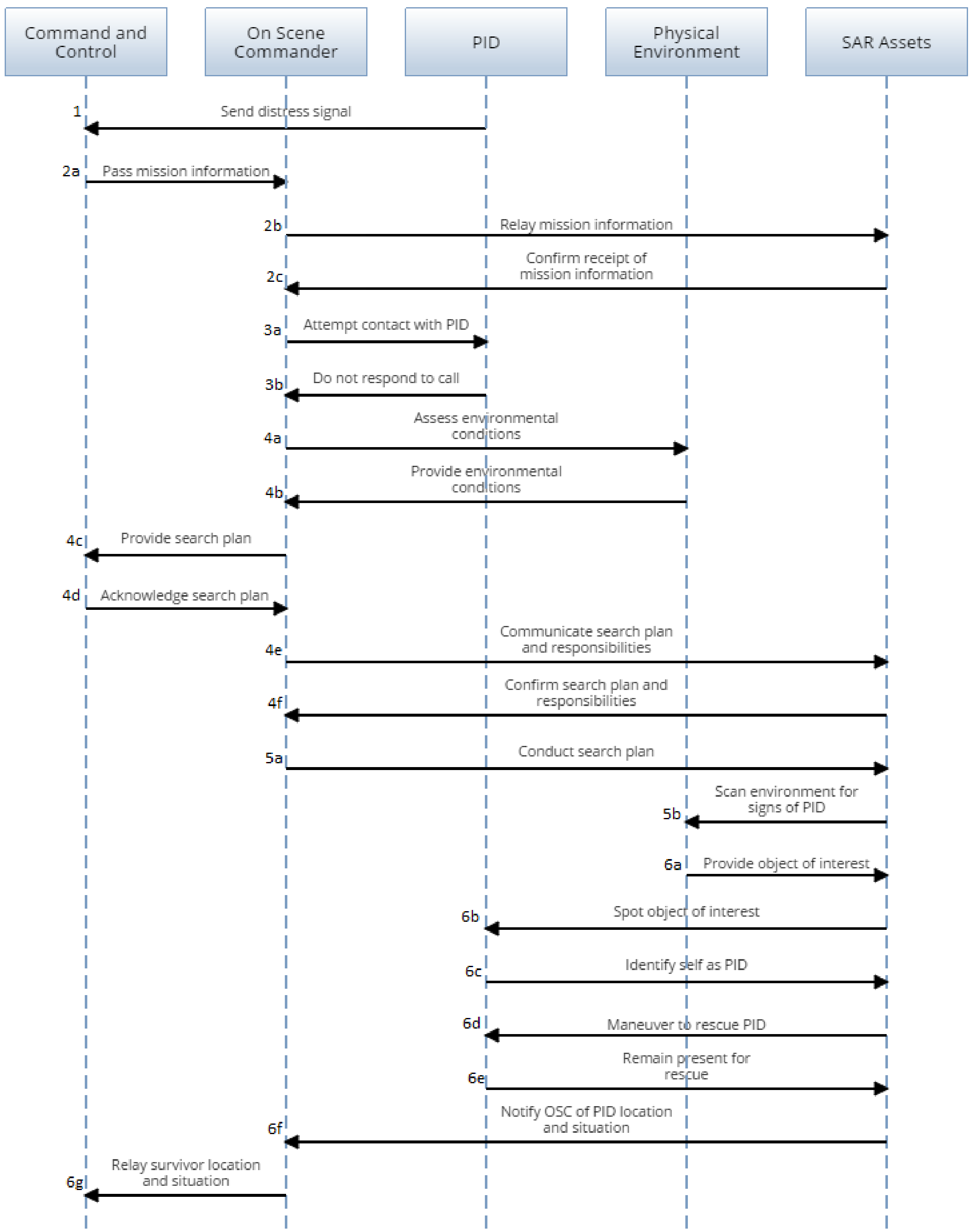

Figure 2. SAR DRM Sequence Diagram

Based on the SAR background study and the DRM, the five major assets interacting in the Figure 2 sequence model are C2, OSC, PID, Physical Environment and

SAR Assets. C2 and the OSC have been discussed in detail regarding their purposes in the SAR system. The PID is representative of any person or groups of people that are in need of SAR assistance. The PID asset is essential to the model because it initiates the response from the SAR system and is the reason for the mission. The Physical Environment asset pertains to the SAR mission space and has important interactions with the PID, OSC and SAR Assets. The PID interacts with the environment while performing survival activities and that encompasses anything the PID does in order to stay alive. Since the PID can only do this for a finite amount of time, this interaction can significantly affect the mission depending on the environmental conditions. The environment also dictates some of the capabilities and procedures of the OSC and SAR Assets. Search plans are necessarily predicated off factors like sea state, visibility, time of day, and other weather factors, thus the inclusion of the Physical Environment asset. Finally, SAR Assets pertain to any vessel in the mission space that contributes to the SAR system. These vessels need not have direct ties to C2 or the mission itself, but if they are in the area and could potentially be of assistance, they fall under the SAR Asset entity.

Moving down from the five assets at the top, the horizontal messages are sequentially ordered and connect the asset lifelines according to the mission narrative and general rules outlined in Chapter IV. Each message is numbered to the left of its horizontal line to indicate its relationship to a section of the mission narrative. Recall that the narrative is broken into six separate paragraphs that describe the flow of the mission and correspondingly, Figure 2 contains number labels from one to six. The purpose of the letters is to show sequence order within the major narrative paragraphs if multiple steps are performed. This numbering convention is not required for a sequence diagram, but it is helpful in tracing the steps of the sequence diagram to the words of the mission narrative. This is one way to ensure that there are no discrepancies between the narrative and the model. Additionally, the numbering makes organization easier as multiple paths are analyzed and can even aid in tracing events through multiple models in order to validate work completed across multiple techniques.

2. Model Assessment

The major strengths of the sequence diagram are its simplicity and intuitive organization. There is nothing simple about a SAR system architecture and its interactions, yet the sequence diagram has provided a means of assessing the major assets and their roles in the system as they pertain to the mission narrative. The diagram is straightforward to view, logically organized, and easily traceable to the base narrative. Additionally, the diagram is simple to learn and easy to build and manipulate. For example, if a deviation occurred at step 3b whereby the PID does respond to the OSC and gives a precise location, then the OSC could proceed directly to that position or send a SAR asset to make the rescue. This would eliminate the need for a search plan so steps 4c through 6c could be eliminated and thus a new sequence diagram would be created for the specific instance of the PID responding to the OSC's call. Changes need not come only from manipulating individual steps in the narrative. The general rules also provide a means to change the mission scenario if, for instance, the OSC has to return to base due to a malfunction or low fuel. In this situation, a new step would be required after 5b whereby the OSC communicates its issue and intention and then additional interactions would be needed by C2 in order to dispatch another OSC asset to the scene to continue with the mission. Of note, it is by no means necessary to only change one step at a time for any particular mission path. The "if-then" nature of the narrative and general rules are specifically designed to allow for such flexibility in the model.

Another major strength of the sequence diagram is its ability to be translated into other models. Figure 2 was constructed using Innoslate, which allows models to be translated into other forms and techniques as long as certain constructs are obeyed when creating the original diagram. Even if that were not the case, a pen and paper version of Figure 2 could easily translate into many of the models investigated by this study. This idea will be explored further as other models are presented and the validity of this assertion is further proved through a model consistency trace contained in Appendix A.

Something to consider when choosing to use a model such as the sequence diagram is the general exclusion of functions resulting in interactions that do not occur directly between the assets. The construct of the sequence diagram is focused on interactions between the assets and as a result, does not encompass anything that an asset might do in the mission that has no effect or causal relationship with another asset. For instance, step 1 has the PID sending a distress signal that is received by C2. The mission narrative states that a number of things go on at C2 after receiving that distress signal before they would ever reach step 2a of passing mission information on to the OSC. Since the C2 internal action cannot be accounted for in the sequence diagram, there is also no way to assign a time value to it for any kind of executable modeling. A similar exclusion occurs with the OSC between steps 1 and 2a as well. Presumably, C2 would give some kind of launch order for the OSC once they accept the mission, but there is no indication of any kind of departure from base for the OSC or any other SAR assets. Although other tools can show these activities as non-interacting bars on the lifelines, the strength of sequence diagrams is showing interactions between assets. The end result is that the sequence diagram necessitates some abstraction when actions by individual assets have no interactions with other asset lifelines in the diagram. This means that sequence diagrams could have some accuracy limitations in executable simulations that may be desired, and it could lead to some confusion in the absence of a good mission narrative because intuitive leaps are required to connect the messages in the sequence.

Another consideration for the using the sequence diagram is the time required to construct multiple models. In Innoslate, there is no method to generate multiple sequence diagrams based on alternative paths that may arise in the SAR mission. This limitation is most relevant for modeling that seeks to analyze many different scenario paths or multiple asset interactions within a system. For something as dynamic as a SAR scenario, the possibility exists for hundreds of iterations for a single mission narrative. If a modeler then desires to consider multiple OPSITs, the number of iterations could easily push into the thousands. Constructing that many models without error is obviously untenable, so if modelers wish to build multiple paths into a collection of sequence diagrams, they must find another way to efficiently generate the models.

A final point on constructing sequence diagrams in Innoslate that is important to mention is the difference between the two modes of operation. Figure 2 was constructed in what is known as parallel mode. This mode is the most powerful option because it creates sequence diagrams that generate parallel “swim lanes” for each asset lifeline so that the sequence diagram can be translated into an action or activity diagram. In terms of constructing a proper parallel mode sequence diagram, this means that all messages between the assets must have a predecessor message for continuity. For instance, step 4c in Figure 2 has the OSC providing a search plan to C2 and C2 acknowledges the search plan back to the OSC in step 4d. Parallel mode requires a return message from C2 so that the OSC can complete step 4e. Without it, the program is forced to fill in what it considers a continuity gap if step 4d was omitted and the OSC simply communicated the search plan to the SAR Assets right after providing the search plan to C2. When that happens, the executable model does not properly reflect the order of the sequence diagram and provides incorrect analysis. The same is true of step 3b where the PID does not respond to the OSC’s call. It might seem unnecessary to model a non-interaction but omitting it in this case produces another continuity gap in the sequence. If greater flexibility is desired, sequence mode generates diagrams based purely on sequential action and activity diagrams without the parallel “swim lanes.” Thus, there is no need for the response message from each asset that has an interaction. The consequence of this flexibility is far less automation in assisting with the construction of an executable model. The translation construct will be discussed in further detail in this chapter’s section on executable models.

3. Insights

It is important to note that the order in which the various modeling techniques are presented in this chapter does not necessarily reflect the order in which they were completed. Chapter II outlined that the modeling effort would undergo multiple iterations with each technique based upon insights that were gained throughout the process. The sequence diagram was constructed first, but that does not mean that it did not undergo refinement. For instance, one of the difficulties encountered while constructing the various models was what to do with objects of interest spotted in the mission space. Figure 2 shows a message from the SAR Assets to the PID in step 6b indicating that the object of interest in the ideal case is the PID and so the mission progresses to the rescue after a PID is identified. While constructing other models, the question quickly arose for alternate paths of what the interaction would look like if the Object of Interest was not related to the SAR mission. Clearly, the message in 6b could no longer point to the PID lifeline in this case and so a sixth “Object of Interest” asset was added to the top of the sequence diagram. This construct initially reconciled the situation where a spotted Object of Interest was not related to the SAR mission. Ultimately, the asset for the Object of Interest was removed as its own entity in favor of the final product of Figure 2. In the situation where an Object of Interest provided by the environment is not related to the SAR mission, the messages following step 6a can occur between SAR Assets and Physical Environment with no need for an extraneous entity with its own lifeline. Just as the original idea to add the Object of Interest asset did not come solely from analyzing the sequence diagram, the idea to remove it occurred through multiple iterations of trying to work the idea and interaction across different models. This is why it was so important to construct and iterate all the models simultaneously according to the methodology presented in Chapter II.

The decision to start with a sequence diagram versus some other modeling construct when analyzing a system is wholly dependent on what the system is and what information is available at the onset of the modeling. Since the SAR problem has a natural flow from the background study to the DRM, the interactions between the assets were modeled via sequence diagram first because the mission narrative was available and easily translated. The assets were already identified in the background study so all that remained was connecting messages to the asset lifelines in accordance with the mission narrative. In a situation where less information is known about specific system interactions between assets, it might be more desirable to begin with a model that allows for more abstraction than a sequence diagram. That way, the individual interactions could be developed further through other means of analysis before an attempt is made to connect the assets to each other via messages in a sequence diagram.

B. IDEF0

Integration Definition for Function Modeling (IDEF0) is a technique used to model the decisions, interactions, and activities of an organization or system. It evolved from the Structured Analysis and Design Technique (SADT) graphical language when the United States Air Force commissioned the developers of SADT to design a function modeling technique for analyzing and communicating a system from a functional perspective (Knowledge Based Systems Incorporated, 2010). When used correctly, IDEF0 models simplify and organize the analysis of a system. Since IDEF0 is capable of graphically representing a wide variety of operations to any desired level of detail, it promotes consistency in interpretation not only between analysts and designers, but also to the customer. Ultimately, IDEF0 is intended to assist in identifying what functions are performed by a system, what is needed to perform them and how to improve the design based on what is right and wrong with the system. As a result, IDEF0 models are often a starting point for systems modeling efforts.

The two primary components of an IDEF0 model are functions, represented in the model by boxes, and data or objects that relate the functions to each other, represented in the model by arrows (DOD Systems Management College 2001, 51). The arrows connecting the functions come in one of four categories, which are called inputs, controls, outputs and mechanisms. Functions, governed by controls, transform inputs into outputs and those functions are performed by mechanisms. In the model, inputs always enter the function box from the left, controls always enter from the top, mechanisms are always positioned at the bottom, and outputs always leave from the right. Inputs, controls, outputs and mechanisms each have a specific definition according to the IDEF0 standard and those definitions are provided in Table 3 below. Figure 3 follows the table and shows the basic construction of a function along with input, control, output and mechanism arrows.

Table 3. IDEF0 Terms (from DOD Systems Management College 2001, 51)

- **Function**: A transformation of inputs to outputs, by means of some mechanisms, and subject to certain controls, that is identified by a function name and modeled by a box.

- **Input:** In an IDEF0 model, that which is transformed by a function into an output.

- **Output:** In an IDEF0 model, that which is produced by a function.

- **Control:** In an IDEF0 model, a condition of set of conditions required for a function to produce correct output.

- **Mechanism:** In an IDEF0 model, the means used by a function to transform input into output.

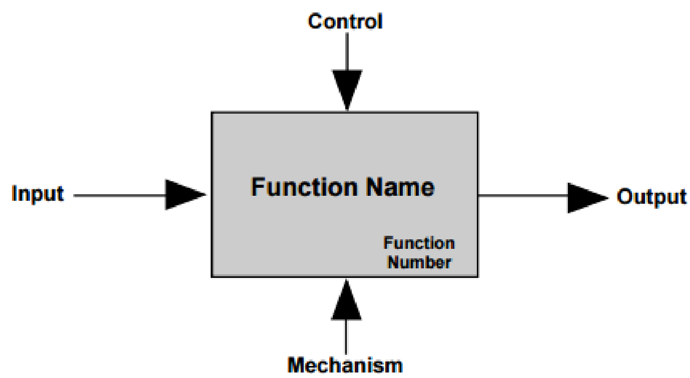

Figure 3. IDEF0 Format (from DOD Systems Management College 2001, 51)

The IDEF0 modeling process starts by identifying the major system function for decomposition. This prime function is then portrayed on an IDEF0 context diagram that shows the interactions between the system's mechanisms at the highest level. From this context diagram, lower level diagrams are generated by further decomposing the function of focus into its individual functions, along with all of the corresponding inputs, controls, outputs and mechanisms. There is no limit to how many times decomposition can occur for a specific function, as it all depends on how much of a detailed breakdown is desired for any particular design.

When performing modeling with IDEF0, it is important to remember that unlike sequence diagrams and other use case diagrams, IDEF0 models do not show a sequenced flow of interactions. The point of IDEF0 is to show a general view of information and resource flow among functions and entities occurring at a high level of abstraction and as such, modelers and interpreters should not expect any logical execution of activities in a time-sequenced order (Giammarco, Hunt, and Whitcomb 2015). Furthermore, since IDEF0 models are more abstract than use case models, it is desirable to name the inputs, controls, outputs and mechanisms abstractly so that they can be relevant for a variety of more specific instances of interactions among system entities and functions. The reason for doing this is simplification and to keep clutter to a minimum on the higher-level diagrams. Systems are complex, which is why tools like IDEF0 exist to decompose them, and trying to show every possible detail on the higher-level diagrams would be overwhelming and thereby negate the usefulness of the model. This is an important concept to adhere to throughout IDEF0 decompositions. If more detail is required, then the answer is to decompose the function into another level of IDEF0 rather than risk a useless diagram that is overly complicated and therefore cannot be read.

1. Model

Figure 4 is the IDEF0 context diagram for the SAR DRM. At first glance, it seems like a daunting visual representation, but breaking it down by component shows the simplicity and relative ease of use. Starting with the mechanisms at the bottom of the diagram, it is apparent that these entities are the same five assets from the sequence diagram. These five assets are the embodiment of the SAR system from the DRM. In the IDEF0 context, all mechanisms perform the functions in the boxes above the ones to which they are connected. Each mechanism has a specific function box and the name of each function is sufficiently broad and abstract to encompass every interaction the individual mechanism could have for any subsequent decompositions. For example, the PID function is Perform Survival Activities. This function encompasses everything the PID would do in any situation in order to stay alive in the environment.

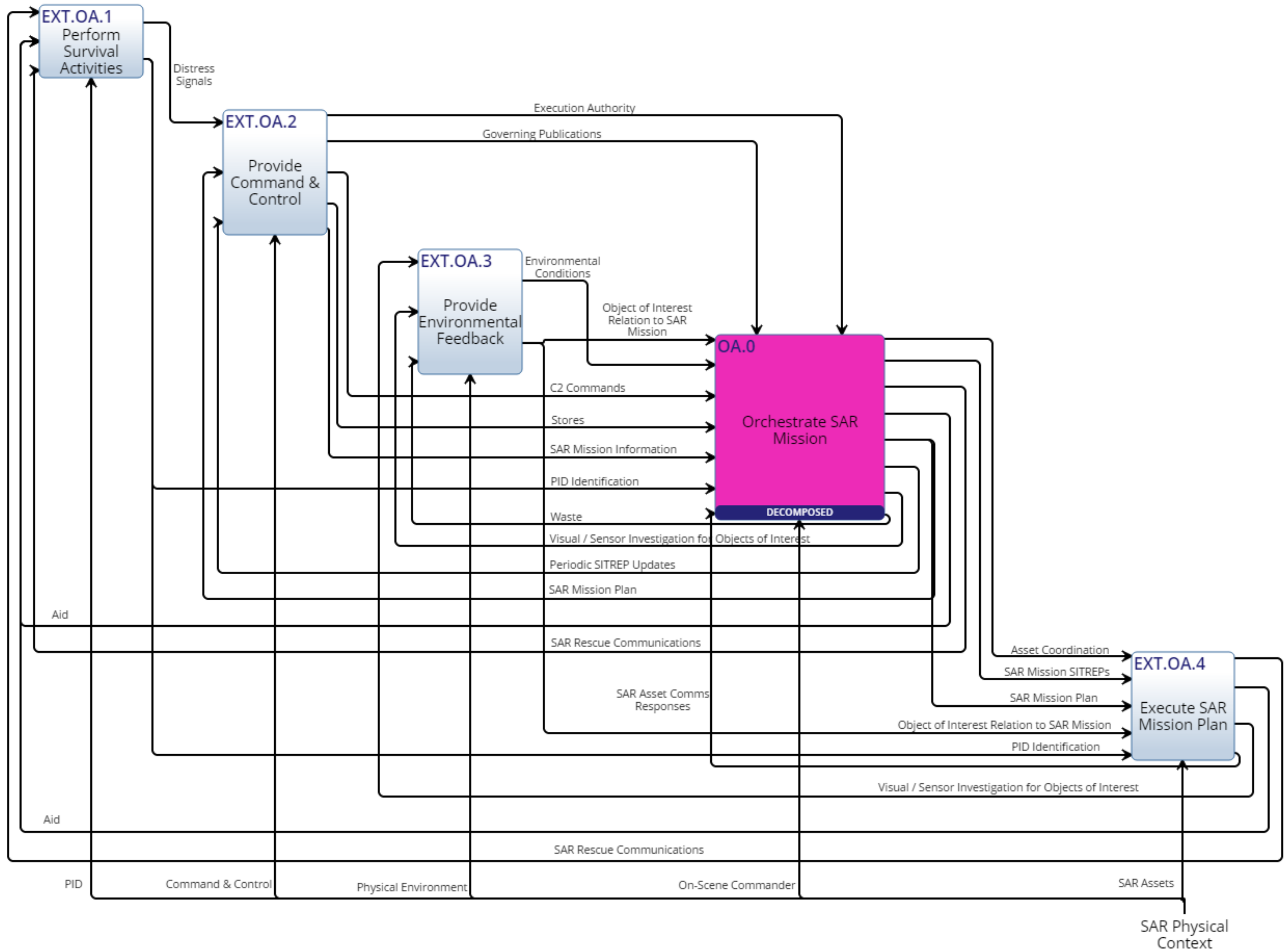

Figure 4. SAR DRM IDEF0 Context Diagram

If the PID function block were decomposed, the next level down could investigate relationships with survival activities over water, in the mountains, in the desert or anywhere else. This is why the context IDEF0 must necessarily be abstract and general, because all of that information could not possibly fit into Figure 4. Since the OSC is the focus of the modeling for this study, its number is OA.0, which indicates that it is the function of interest with which all of the external operational activities numbered one through four interact. The OSC function box is differently colored to make it stand out, although this is not a requirement, and the decomposed ellipse at the bottom of the box indicates that another level of decomposition exists for that function below the context diagram.

The next thing to notice in the SAR DRM context IDEF0 is the arrows for inputs, outputs and controls. Unlike the sequence diagram, the arrows connecting the functions in IDEF0 are not action statements and once again, they are sufficiently abstract so that they can be carried forward to lower-level diagrams in multiple instances where they may occur. Since the OSC has a lower-level decomposition, its inputs, outputs and controls are pointed out specifically herein. To begin, the OSC mission function has two control arrows coming from C2 in the form of governing publications and execution authority. This indicates that the OSC operates under the authority of C2, but also adheres to governing publications that could include SAR manuals, aircraft operating manuals, and even individual unit standard operating procedures. OA.0 also receives input from the other assets' functions. Inputs from C2 include commands, stores (fuel, cargo, SAR equipment) and SAR mission information. Inputs from the environment include environmental conditions and the object of interest and its relation to the SAR mission. From the PID, the OSC receives an input for PID identification and from the SAR assets, the OSC function receives an input for all SAR asset communications responses. For outputs, the OSC generates asset coordination and mission SITREPs that go as inputs to the SAR assets, and a SAR mission plan that goes as an input for C2 and the SAR assets. There are also rescue communications and aid outputs that go as inputs to the PID, and periodic SITREP updates back to C2. Finally, the OSC generates output waste and a visual or sensor investigation for objects of interest, both of which go to the environment as inputs. Although it is not intuitive to step from the DRM mission narrative directly to the IDEF0 context diagram in Figure 4, it is apparent that components of both the narrative and the sequence diagram are encompassed by the abstract relationships presented between the high-level asset functions. Together, these functions and their relationships make up the SAR physical context, which is annotated on the lower right portion of the diagram. Finally, it is important to note all of the arrows touching OA.0 because they will be seen again and must be accounted for in the decomposition of the Orchestrate SAR Mission function.

Just like every function on an IDEF0 can be decomposed into constituent sub-functions, the inputs, controls, outputs and mechanisms are also decomposed on lower-level diagrams. In IDEF0 terminology, this decomposition process is known as stepwise refinement, and provides the means to increase the level of detail in the “child” diagrams that was not possible on the higher-level “parent” diagrams (Giammarco, Hunt, and Whitcomb 2015). Figure 5 presents such a decomposition for the OSC function OA.0 Orchestrate SAR Mission. Since OA.0 is considered the topmost system function for this study, the decomposition in Figure 5 is considered a first level IDEF0 diagram.

On the first level IDEF0 for OA.0, the focus shifts specifically to the OSC’s role in accomplishing the function of orchestrating the SAR mission. As a result, the mechanisms at the bottom of Figure 5 are now the individual components of the OSC entity vice the five major SAR system assets from the context diagram. Notice that the mechanisms themselves are general so that they can be applicable whether the OSC is a helicopter, a ship, an airplane, a jeep, or even an unmanned vehicle. It has been stated previously that the relationships between functions should be labeled abstractly so that they can easily be decomposed. It now becomes clear that mechanisms on various levels should also include abstraction so that an IDEF0 can be applicable across multiple systems or to assets that could accomplish the same function up in the context diagram.

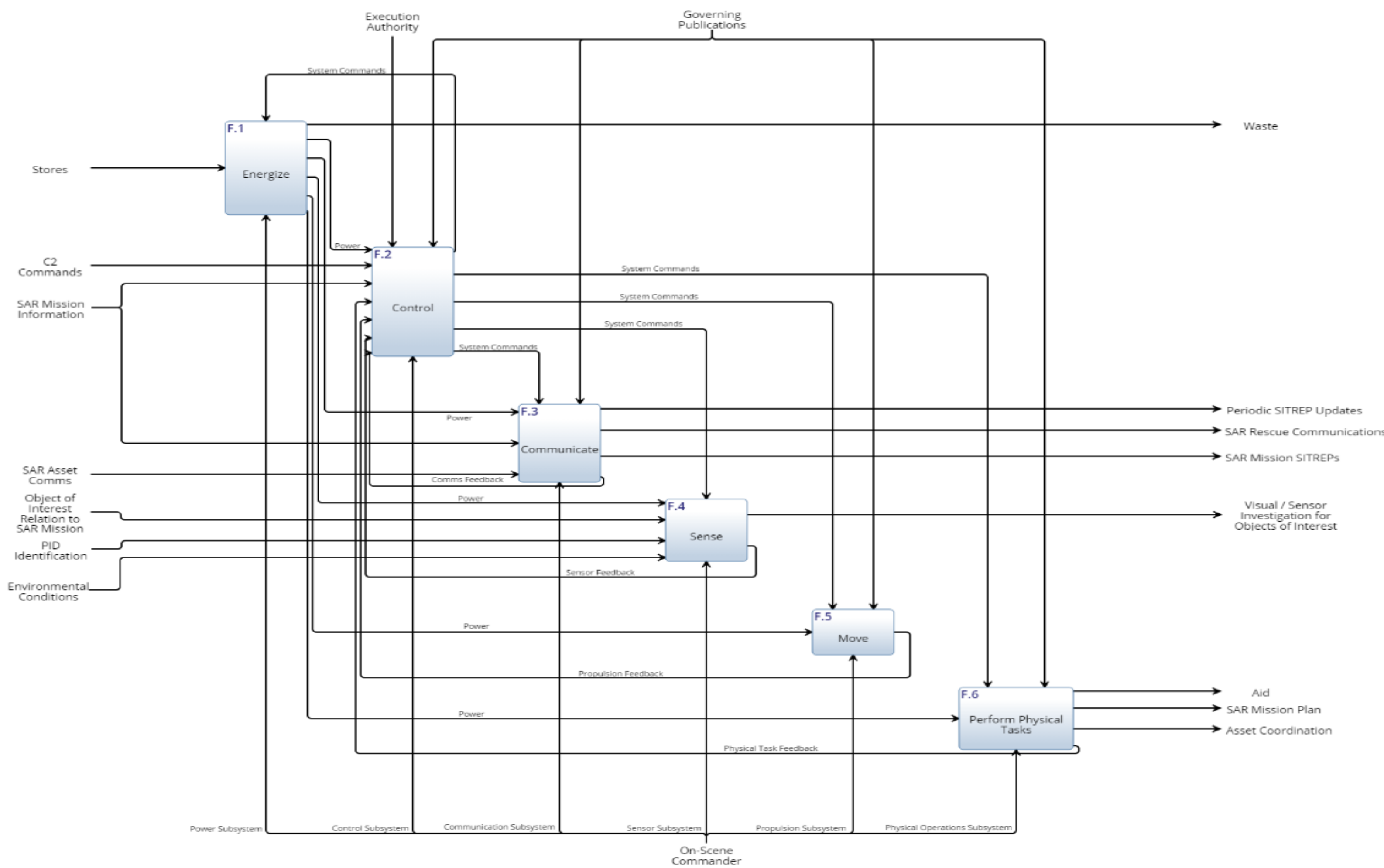

Figure 5. SAR DRM IDEF0 OA.0 Decomposition (after Giammarco, Hunt, and Whitcomb 2015)

The functions corresponding to the first level mechanisms are action-oriented and describe what each mechanism does for the OSC—the power subsystem energizes, the propulsion subsystem provides movement, the control subsystem controls, the communications subsystem communicates, the sensor subsystem senses, and the physical operations subsystem provides the means to carry out physical tasks. This last function includes anything on the OSC used to accomplish tasks that are not already encompassed by the other five functions. For instance, it could be a rescue hoist system on a helicopter or a crane winch on a ship. Options for more detail on any of the six functions would occur on a second-level diagram if further decomposition were desired.

Located about the periphery of the first level diagram are all of the inputs, outputs and controls that touched OA.0 up in the context diagram. The first level diagram allows those relationships to be assigned to specific functions within the OSC asset itself. This way, a modeler can see subsystem breakouts in order to ascertain which components perform what functions in accomplishing the mission from the context IDEF0. This is the amount of detail that is possible as decompositions occur on the lower levels. From a notational standpoint, the OSC relationships from the context diagram originate on the outer perimeter of the first level diagram and either go into the first level functions as inputs and controls, or exit as outputs. The reason for originating on the outside of the diagram is to separate those relationships from ones generated from the functions and mechanisms on the first level. That way, there is no confusion about where SAR Mission Information came from as an input to F.3 Communicate and how it differs from a Comms Feedback output on F.3 versus a SAR Mission SITREP output on the same function. If F.3 were decomposed further, all of the relationships from the first level would once again be carried down in order to see how the mechanisms of the communications subsystem handle all of the inputs, outputs and controls from Figure 5. The IDEF0 construct requires that these relationships be carried down for continuity because each decomposition must necessarily be an abstraction of its parent diagram. This adds traceability and therefore validity throughout an IDEF0 set of models and ensures that important relationships are not lost as the decompositions become more detailed.

Ultimately, an IDEF0 like Figure 5 is useful for anything from understanding relationships among a system's various mechanisms and functions to determining design requirements. In terms of the SAR DRM, the context diagram has shown what the SAR system as a whole must do to accomplish the mission, while the first level diagram has shown what the OSC must do to accomplish its role in that mission. The OSC function was decomposed because that asset is the focus of this study's modeling, but any of the five major assets from the context diagram could be decomposed in order to evaluate their individual roles in the mission.

2. Model Assessment

The primary strength of IDEF0 is its ability to detail system activities for functional modeling through abstraction. This can be particularly useful when top-level requirements exist and a modeler wishes to determine how to accomplish those requirements through a system's functions and mechanisms. This was the case with the SAR DRM because the mission narrative provided all the necessary information to construct the IDEF0 context diagram and from there, the OSC's function could be broken down into its constituent functions and mechanisms. This is an example of a top-down model analysis, but it is not the only way to use the IDEF0 construct. Suppose that a modeler wishes to evaluate the inclusion of an unmanned vehicle into the SAR system acting as the OSC. In this case, it may not be feasible to start at the top-level context diagram but instead begin at a detailed lower-level IDEF0 for the specific unmanned vehicle. That way, the modeler could group the capabilities of the asset together to ascertain a hierarchy of activities to be carried up to the next level. This recursive process could then carry the asset all the way up to a high-level context diagram where its observed activities can be described and combined into a higher level activity. Perhaps an unmanned vehicle's capabilities as OSC would lead into the same orchestration of the SAR mission described in Figure 4, or perhaps its function would be something entirely different. Regardless, this is the kind of flexible functional modeling that is possible using IDEF0, and it is a major reason that this type of modeling is frequently used at the beginning of many modeling efforts.

The flexibility of IDEF0 is not without its pitfalls, however. Since a major concept when constructing IDEF0 models is concision through abstraction, there is a tendency to overdo the brevity in labeling relationships to the point that interpretation is difficult for any readers outside the modeling effort or the subject matter expert arena. As such, a modeler needs to ensure that relationship labels on the inputs, controls, outputs and mechanisms are sufficiently simple, yet descriptive, to reduce misinterpretation and ease translation for other designers and ultimately the customer. It is also important for the modeler and the reader to avoid interpreting IDEF0 models as representing a sequence of activities. Although IDEF0 was never intended for activity sequences, it is easy to misunderstand this because of the left-to-right nature of inputs and outputs on the diagrams and the fact that functions can be placed sequentially left-to-right and top-to-bottom so that outputs from one function go as inputs into the next function. Obviously, for more complex systems such as the SAR system, the numerous feedback loops between the various functions make a sequential interpretation more difficult on the higher level diagrams. Nevertheless, any IDEF0 model should avoid embedding unintentional sequences whenever possible.

3. Insights

The IDEF0 models were constructed in the middle of this study's modeling effort, and for the SAR DRM, IDEF0 was particularly helpful in analyzing the relationships between the top-level asset mechanisms and their functions. Before Figure 4 and 5 were completed, a lot of time was spent on sequenced use cases and moving to the IDEF0 provided an opportunity to see what relationships were occurring beyond the sequenced steps of the mission narrative. Obviously the intent of the general rules was to fill in some of the intuitive gaps inherent with a stand-alone narrative, but now with IDEF0, there is an opportunity to observe the recurring SITREP updates, rescue communications and C2 commands in the context diagram. Beyond the general rules, the context diagram also allows for depictions such as the governing publications and execution authority controls that guide the OSC in orchestrating the mission, and even the input of stores to the OSC (such as smoke markers, fuel, or rafts) along with a waste output to the environment. Additionally, whereas the sequenced cases only allow for one specific iteration of the narrative at a time, the abstract nature of IDEF0 allows the model to show many potential narrative outcomes. This is highlighted by the aid outputs coming from the OSC's function as well as the SAR Assets. The sequence cases can only show one unit helping for any particular case, but IDEF0 presents the more realistic situation where aid could come from any unit on scene. A similar interaction is present with the PID identification relationship, which could also occur with any unit on scene.

The IDEF0 model also aided in validating the decision to remove the object of interest from the group of major assets in the SAR system. Like the sequence diagram, the first iteration of the context diagram contained six mechanisms and six EXT.OA functions. When the decision was made to remove the object of interest as its own asset, looking back at the context IDEF0 proved that there were indeed extraneous relationships acting as inputs and outputs when the object of interest was its own mechanism with a separate function. Even the function name for the object of interest—exist in mission area—was extraneous because intuitively, every mission space will contain objects of interest requiring investigation so there is no need to have a completely separate mechanism and function to account for it. Eliminating that mechanism and placing it under the environment's mechanism and function allowed for a consolidation of relationships on the context diagram whereby a simple two-pronged output from EXT.OA.3 shows the object of interest's relation to the SAR mission to both the OSC and SAR Assets. Any further detail on the objects of interest themselves would be contained in a first level decomposition of EXT.OA.3, which is a much more appropriate place for them to exist as a mechanism than on the high-level context diagram.

Finally, the IDEF0 has shown its usefulness in potentially choosing assets to use as part of the SAR mission. It does this through the decompositions that are possible from the high-level functions. Selecting specific units for tasking as the OSC or SAR Assets is beyond the scope of this study, but the first level decomposition of the OSC in Figure 5 could easily translate into a requirements list for any unit's suitability to orchestrate the SAR mission. Similar decompositions for the SAR Assets function in EXT.OA.4 could be used the same way. Additional decompositions of specific functions on the first level could be used in tradeoff analyses as well. For instance, if there are two choices of vehicle to use for the OSC on a particular mission and their performance characteristics are comparable for all six first level functions except for F.3, then the communicate function would need further decomposition. On the second level diagram, a modeler may find that one vehicle has six radios that can operate on six different frequencies as opposed to three on another vehicle. In this case, the easy choice is the six-radio unit since this unit should be able to communicate with more entities on discrete frequencies to avoid confusion and radio chatter. This example illustrates the ability of the IDEF0 to decompose to minute details, making the IDEF0 a powerful tool for comparison and requirements analysis in addition to observing how mechanisms and functions relate to each other on various levels in the system.

C. HIERARCHY CHARTS AND TREE DIAGRAMS

Hierarchy charts and tree diagrams are organizational charts that show the structure of an organization or system and the relationships and relative ranks of different parts and functions. They were originally designed for charting the organizational structure of businesses but have proven useful for numerous applications in a variety of disciplines such as project management, computer science, system design, and mathematics. In systems engineering, they are akin to a functional decomposition whereby a system’s top-level functions are broken down into their respective sub-functions as a means of ascertaining what functions must exist to accomplish the overall purpose of the system. This type of modeling is not unlike IDEF0 with the concept of decomposing functions, but the execution is far more general than anything contained in the IDEF0 construct. Both hierarchy charts and tree diagrams consist of labeled function nodes, but there is no defined mechanism performing the function as in an IDEF0 model. Furthermore, there are no descriptive inputs, outputs or controls for any of the functions. Each sub-function is simply a decomposition of the task above it. This general construct also allows for physical hierarchies that have no functions at all.

The diagrams themselves usually begin with some version of a root node, which could be an overarching function or simply the name of the system undergoing decomposition. Underneath the root note, branches are constructed for the major functions on the first level, all of which support or make up the function or system at the top. Functions on the same level are considered “peer” functions and each can be decomposed further down depending on the level of detail desired. The idea of the decompositions is to label functions generally in order to keep the diagram as solution-neutral as possible. The further down the decomposition goes, the more difficult this task becomes because minute details usually require function names that contain specific solutions.

Ultimately, the purpose of these diagrams is to gain insight into the identity of constituent components or functions, and to obtain a compressed view of the entire system’s interactions. This is extremely helpful in understanding simple and complex systems because as major functions and components are broken down, it is easier to comprehend the smaller parts that make up the whole. Furthermore, these views simplify the task of figuring out what functions or components must exist to accomplish the higher-level pieces in the hierarchy as designers seek improvement in various components.

1. Model

Figure 6 contains a hierarchy diagram for the SAR DRM activity context. This diagram was constructed with Innoslate and because the program stores the information from the IDEF0, this model view carries over the same function labels and numbering that were used in Figure 4 and 5. This feature is convenient when using a program like Innoslate because it is important to maintain the same functions and numbering when moving from one type of model to another. Sloppy naming and numbering conventions affect the translation of a model from one technique to another and also degrade the accuracy and validity of the modeling effort. As such, the top box in Figure 6, SAR Activity Context, is the same overarching activity from the bottom of the Figure 4 IDEF0. Therefore, on the first level of the hierarchy lie all of the external operational activities along with the operational activity of focus, orchestrating the SAR mission.

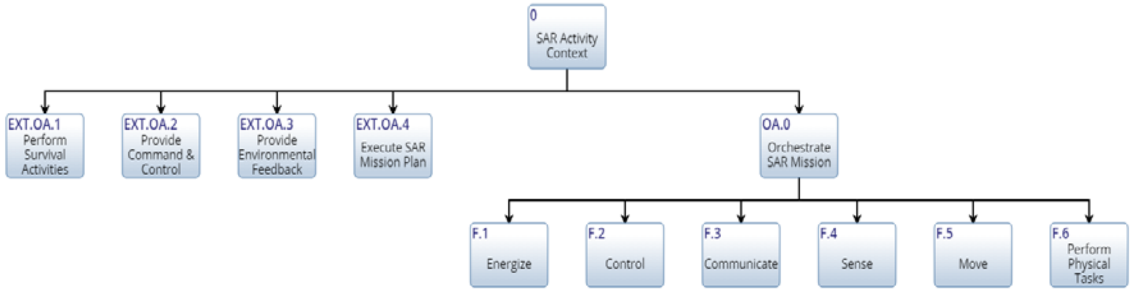

Figure 6. SAR DRM Activity Context Hierarchy Diagram

Since OA.0 has a decomposition, there is another level underneath it containing the six functions from the Figure 5 first level IDEF0 diagram. In this view, all of the functions on the first level are "peer" functions and are the major actions that make up the SAR activity context. F.1 through F.6 are also "peer" functions on the same level, the only difference being that they make up the activity of orchestrating the SAR mission. Of note, there are no longer any specific mechanisms present performing the functions in the hierarchy diagram. This makes the diagram much more solution-neutral than an IDEF0 since there is no performance component.

Figure 7 is a tree diagram illustrating how a hierarchical diagram decomposes the physical aspect of a system. The tree diagram accomplishes a similar purpose as Figure 6, except now the focus is on breaking down physical components vice functions. The same principles apply for the tree diagram view, as components in line with each other are still "peers." Then, the OSC is broken down further into the six subsystems that comprise the OSC unit. Thus, the general nature of the hierarchy construct allows for flexible decomposition activities regardless of the system area of focus.

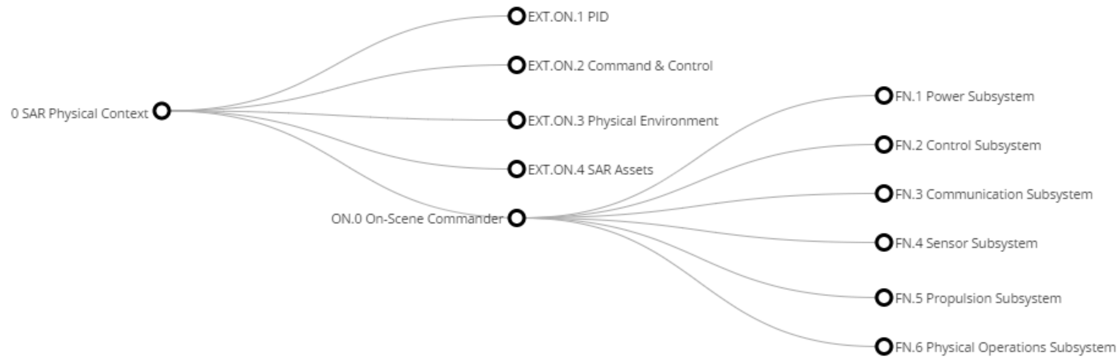

Figure 7. SAR DRM Physical Context Tree Diagram

2. Model Assessment

The strength of any type of decomposition model is that it simplifies complex systems and tasks by breaking them down. This is important because if designers do not understand what a system does or what it needs to do, then it is impossible to meet the requirements of the stakeholders. Throughout the decomposition, these types of models allow the modeler to remain generic and solution-neutral so that interfaces between nodes can easily be updated and replaced as iterations progress. Due to these flexible qualities, a decomposition-type diagram such as a hierarchy or tree diagram is a great place for designers to start if they are having trouble figuring out how to attack a particular problem with system design. An obvious need may exist, but an obvious solution is rarely available. Take OA.0 for instance. Orchestrating the SAR mission is an obvious need in the SAR system, but how can anyone understand what it entails without breaking the function down further? Figure 6 and 7 begin the breakdown process by looking at all the high-level functions required of whatever entity will be responsible for orchestrating the mission. F.1 through F.6 seem straightforward but breaking each of them down will reveal how complex each one really is. Energize, for example, is simply powering the asset but then considerations quickly arise for batteries, generators, alternators, current types, wiring, failure identification and redundant backup systems. These are all functions underneath energize that each have their own breakdowns. That is why hierarchical views are useful; they illuminate possibilities and depict them graphically so that an exhaustive effort can ensue to find the best system design.

Another helpful aspect of hierarchy and tree diagrams is the condensed view of all the functions or components they provide. Consider that it took two large, separate IDEF0 diagrams in order to display an activity context and one level of decomposition versus the single diagram that encompassed both sets of functions in Figure 6 and 7. Although the IDEF0 provides significantly more detail on each of those levels, it is often necessary to provide a single diagram snapshot where all of the functions or components and their decompositions can be viewed. When multiple decompositions are performed in complex systems, there could easily be 20 different IDEF0 views, so having a condensed version with all the basic breakdowns in one diagram can be very helpful. In this way, a modeler can reference a particular function or component and then use an IDEF0 or some other model if more detail is desired.

Something important to consider when constructing a hierarchy or tree diagram is ensuring consistency with the naming convention of the breakdown. A modeler could choose to decompose functions or components but the two cannot be mixed. As is the case for Figure 6 and 7, the functions retain their action-oriented labels throughout and do not switch from functions to components or mechanisms as the levels move down. Labeling the nodes as pure functions or pure components preserves the generic and flexible nature of the diagrams and keeps the model consistent. Otherwise, the model can quickly become confusing.

3. Insights

Given that the hierarchy and tree diagram were constructed after the IDEF0 models, it is tempting to brush them aside in favor of the more detailed models. However, Figure 6 and 7 have more information to offer since in their current form, they only represent a basic decomposition of the model for the SAR DRM. Many further decompositions are possible with F.1 through F.6 to say nothing about taking one of the other EXT.OA functions for decomposition. The point is that there is much more to be learned beyond the basic models presented here, even though such an exploration is beyond the scope of this study. That being said, the major insight gained from constructing the tree and hierarchy diagrams occurred in a somewhat unexpected manner.

It is fair to say that in their current state, the hierarchy and tree diagram do not necessarily provide any extra information because of the detailed nature of the IDEF0 diagrams that were already presented. However, if a modeler wished to further decompose any of the functions in Figure 6 or 7, it is likely that the decomposition would need to occur in the hierarchical view before proceeding to anything more detailed. For example, in decomposing F.6 Perform Physical Tasks, unless a designer has a clear idea of the direction needed in order to accomplish that function, there must be significant time allotted for brainstorming and iteration as the task is broken down. Regardless of the level of abstraction, brainstorming functional breakdowns using IDEF0 is not nearly as straightforward as it is using hierarchy-type diagrams. There are simply too many complex relationships to consider when linking mechanisms to their functions and while this level of detail is necessary for a modeling effort eventually, it may not be the best place to start.

In terms of sequence, hierarchical breakdowns are a great starting point for any modeling effort because they break down complex systems and functional requirements so that they can be easily understood. The generic nature of the models allows for maximum flexibility and brainstorming early so that multiple possibilities can be identified and pursued as required. Since these breakdowns are so closely related to more detailed views like IDEF0, they can then be translated once more is known about specific mechanisms, inputs, outputs and controls. Model-based systems engineering is an iterative process, so a hierarchical breakdown can always be revisited throughout a modeling effort in order to gain further insight or perhaps to take on a new direction entirely

D. MONTEREY PHOENIX

Monterey Phoenix (MP) is "a behavioral model for system and software architecture specification based on event traces" (Farah-Stapleton and Auguston 2013, 271). Its purpose is to capture behaviors and interactions between parts of a system and the environment with which it operates, and it does this through automatic generation of use cases (Farah-Stapleton and Auguston 2013, 271). MP was developed because of a noticeable trend of inconsistent architecture representations that were counterproductive during inspections and reviews because model development efforts were unrelated, duplicative and were producing unsustainable results. The major theme of MP is that modeled architectures matter and if they are developed and utilized properly, they capture the behavior of not only the system, but the system's surrounding environment as well. Properly developed architecture models provide an organized framework to discuss and iterate design decisions and provide a means of verification early in the design process in order to save time and money on costly mistakes. Furthermore, accurate architectural descriptions establish common ground among stakeholders so that important questions can be addressed on development strategies, evaluation metrics, testing, and integration (Farah-Stapleton and Auguston 2013, 271). Monterey Phoenix is not intended to replace other modeling techniques such as SysML and UML, but instead seeks to complement them and emphasize the necessity of automated tools for architecture model generation. The automatic generation of architecture models will be discussed in detail as the SAR DRM MP models are presented.

Monterey Phoenix works by describing the behavior of a system in terms of an algorithm, which contains a step-by-step collection of activities the system uses to accomplish a task. Since "MP represents an event as an abstraction of an activity, the behavior of a system can be modeled as a set of events with two binary relations defined for them" (Farah-Stapleton and Auguston 2013, 273). The relations are precedence, annotated as PRECEDES in the code lines, and inclusion, annotated as IN. PRECEDES indicates that the first action occurs before the second, which tells MP the sequenced order of events for arrow traces during automatic generation. IN can be thought of as decomposition, meaning that the event on the left of a colon includes everything on the right of a colon. The colon itself is the execution of the IN relation. For example, to decompose the SAR activity context into the top five high-level assets in MP, a code line could read `SAR_Activity_Context: { C2, OSC, PID, SAR Assets, Physical Environment }`. Since the colon executes the IN relation, all five assets in the unordered set on the right would be a decomposition, or inclusion, of the SAR activity context in an MP trace. More discussion on MP code and event traces occurs in the next section where the SAR DRM model is presented.

1. Model

The easiest way to present the SAR DRM MP code and subsequent event traces is to start with the simplest trace and show what pieces of the MP code brought about the graphical representation. When the code is properly understood, the more complex traces are easier to analyze. The first section of code is shown in Figure 8. Line 1 denotes that the first section will contain the actors in the system and line 3 gives a name to the schema, or the set of event traces based on this code. What follows in line 5 is the naming of the first actor, Command_and_Control, which is instigated using the ROOT MP grammar identifier. In this case, ROOT is analogous to placing an asset at the top of a sequence diagram and ensures that C2 will get its own lifeline in the event trace view. To the right on line 5 is a decision relationship (identified by the vertical line between the blue and orange text in parentheses) between performing normal operations and initiating a SAR mission. In all instances where there is no distress signal received by C2, it will remain in its normal operations state. In the event of a distress signal, C2 will initiate the SAR mission. The orange text for Initiate_SAR_mission denotes further action by C2 for an MP event trace, should a distress call arise. Line 8 decomposes the event on line 6 to show what actions will be generated by MP and therefore performed by C2 in initiating the mission. Brackets indicate an optional event that may occur depending on other interactions in any particular event trace. Receive_SITREP_update in line 11 is an example of one of these optional events. All of the other actors in the MP model are coded in a similar fashion along with their respective actions, should the SAR system respond to a distress call. The only outlier is the physical environment because it provides environmental conditions whether a SAR mission is going on or not. The plus signs around the text on line 36 ensure that MP accomplishes a trace with that action one or more times, up to a scope limit set by the user, ensuring in every event trace that the environment provides its conditions. Thus, the environment has no alternative of normalcy like the other four actors.

The code is slightly more complex in lines 40–43 where the physical environment provides an object of interest. Note there are three different decisions where the object is not related to the SAR mission, the object is wreckage, or the object is the PID identifying itself. These decisions come into play for interactions with the SAR Assets. In line 51, the SAR assets have another composite action (orange) underneath Participate_in_SAR_mission. Line 53 is indented to show that SAR_Assets_scan_for_signs_of_PID is included underneath the SAR assets participating in the mission. When an asset spots an object of interest, it automatically assesses it because there is no alternative operator separating those two actions in line 54. If no object is spotted, then the assets continue scanning. Line 57 is further indented to show its relation to the two orange actions above it. Lines 58–59 once again show the three possible outcomes for identifying the object of interest. In this case, however, the nomenclature is slightly different than that of lines 41–43. This is necessary because when an object of interest is provided by the environment, its relation to the SAR mission will first be indicated by the environment and then identified by the SAR assets. This means there will be similar actions in the lifelines for the environment and the SAR assets and they must have different names to portray the appropriate action and avoid confusion. Finally, if the object of interest is identified as the PID, the two concluding actions for the SAR assets are maneuvering to rescue the PID and notifying the OSC of the situation.

```
/* Actors */

SCHEMA WideRangeSearch

ROOT Command_and_Control:   ( Perform_C2_normal_operations |
                              Initiate_SAR_mission );

Initiate_SAR_mission:   Receive_distress_signal
                        Pass_mission_information
                        Acknowledge_search_plan
                        [ Receive_SITREP_update ];

ROOT On_Scene_Commander:    ( Perform_OSC_normal_operations |
                              Lead_SAR_mission );

Lead_SAR_mission:   Receive_mission_information
                    Depart_from_base
                    Relay_mission_information
                    Attempt_contact_with_PID
                    OSC_assesses_environmental_conditions
                    Provide_search_plan
                    Communicate_search_plan_and_responsibilities
                    OSC_scans_for_signs_of_PID
                    (* Receive_updates_for_OSC *)
                    [ Receive_PID_location_and_situation
                    Relay_survivor_location_and_situation ]
                    Return_to_base;

ROOT PID:       (   Is_not_in_distress |
                        Is_in_distress );

Is_in_distress: Send_distress_signal
                [Identify_self_as_PID
                    Remain_present_for_rescue] ;

ROOT Physical_Environment:  ( + Provide_environmental_conditions + );

Provide_environmental_conditions: [ OSC_assesses_environmental_conditions ]
                                  [ OSC_scans_for_signs_of_PID ]
                                  ( * Provide_object_of_interest
                                      ( Indicate_not_related_to_SAR |
                                        Indicate_wreckage |
                                          Identify_self_as_PID ) * );

ROOT SAR_Assets:     ( Perform_SAR_Assets_normal_operations |
                        Participate_in_SAR_mission );

Participate_in_SAR_mission: Proceed_to_mission_area
                            Confirm_receipt_of_mission_information
                            Confirm_search_plan_and_responsibilities
                            ( + SAR_Assets_scan_for_signs_of_PID + );

      SAR_Assets_scan_for_signs_of_PID:
        ( Spot_object_of_interest Assess_object_of_interest |
               Keep_scanning );
            Assess_object_of_interest:
                ( ID_object_as_not_SAR_related |
                    ID_object_as_wreckage | ID_object_as_PID )
                    [ Provide_updates_to_OSC ];

                ID_object_as_PID:   Maneuver_to_rescue_PID
                                    Notify_OSC_of_PID_location_and_situation;
```

Figure 8. Monterey Phoenix SAR DRM Code Lines 1–64

Figure 9 is the event trace generated by MP when no distress call is received by C2 from the PID. Here, every asset is in a normal operational state with no further action required. All of the normal operational states are blue boxes corresponding to the blue text in the MP code lines, except for the physical environment. The gray dashed arrows are the graphical representation of the IN relation, showing that in this trace, MP has stopped at the highest level of the decomposition because there was no distress call from the PID forcing a decision to include anything else from the root actor activities. While this particular trace is quite simple, it displays the functionality of MP automatically generating use cases based on the input code. This view would not be possible in a regular sequence diagram because there are no horizontal relationships to display between the assets since they are not interacting in this trace.

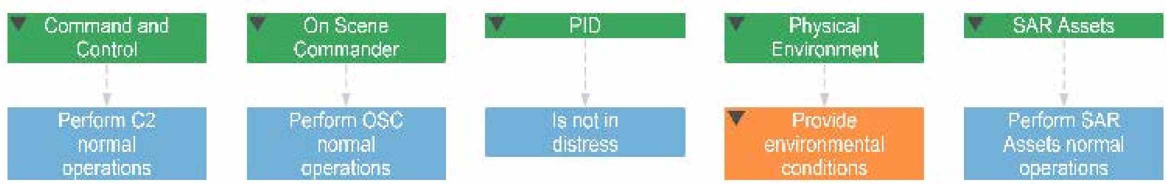

Figure 9. Monterey Phoenix Event Trace: No Signal From PID

Figure 10 shows lines 65–134 of the MP code. This section contains the interactions occurring between the actors and the various decomposed activities that could occur based upon the code from lines 1–64. The interactions are accomplished via the COORDINATE composition, which in conjunction with the PRECEDES relationship, tells MP the order of the events for the trace. The best way to think of COORDINATE is as a synchronous or asynchronous loop over one or more event sets of equal size. For each set of events selected from the coordination sources, MP will perform composition operations between the DO and OD on the code line (Auguston 2015, 8). For example, line 68 identifies the initiating event of the PID sending a distress signal to C2. The $a symbol defines that for this set of events, $a is Send_distress_signal, initiating from the PID. On line 69, $b is defined as Receive_distress_signal, initiating from C2. Then on line 70, $a is given precedence over $b, meaning in this case that each $a happens before a corresponding $b, and the OD at the end of the line closes the coordination operations that began with the DO at the beginning of the line. All of the coordination in Figure 10 is synchronous, meaning that all selected events are totally ordered and have the same number of elements (Auguston 2015, 8). This is consistent on all of the coordination lines as they each contain two elements and are properly ordered with PRECEDES. MP does allow asynchronous coordination for dissimilar numbers of selected events that are not ordered, but that analysis is not applicable for the event traces in this model.

The final piece to cover on the code is the SHARE ALL feature. This accomplishes a shared relationship on the lifelines between identified actors and a specific action. For instance, line 84 has the OSC and Physical Environment sharing the assessment of the environmental conditions. In the more complex traces, this will show a relationship between both lifelines on that action since both actors share involvement in the environmental assessment.

```
/* Interactions */
COORDINATE  $a: Send_distress_signal        FROM PID,
            $b: Receive_distress_signal     FROM Command_and_Control
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Pass_mission_information    FROM Command_and_Control,
            $b: Receive_mission_information FROM On_Scene_Commander
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Relay_mission_information   FROM On_Scene_Commander,
            $b: Proceed_to_mission_area     FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Confirm_receipt_of_mission_information      FROM SAR_Assets,
            $b: Attempt_contact_with_PID                    FROM On_Scene_Commander
                DO ADD $a PRECEDES $b; OD;
On_Scene_Commander, Physical_Environment SHARE ALL OSC_assesses_environmental_conditions;
COORDINATE  $a: Provide_search_plan         FROM On_Scene_Commander,
            $b: Acknowledge_search_plan     FROM Command_and_Control
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Acknowledge_search_plan                         FROM Command_and_Control,
            $b: Communicate_search_plan_and_responsibilities    FROM On_Scene_Commander
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Communicate_search_plan_and_responsibilities    FROM On_Scene_Commander,
            $b: Confirm_search_plan_and_responsibilities        FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
On_Scene_Commander, Physical_Environment SHARE ALL OSC_scans_for_signs_of_PID;
COORDINATE  $a: Provide_object_of_interest  FROM Physical_Environment,
            $b: Spot_object_of_interest     FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Indicate_not_related_to_SAR     FROM Physical_Environment,
            $b: ID_object_as_not_SAR_related    FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Indicate_wreckage   FROM Physical_Environment,
            $b: ID_object_as_wreckage   FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
PID, Physical_Environment SHARE ALL Identify_self_as_PID;
COORDINATE  $a: Identify_self_as_PID    FROM PID,
            $b: ID_object_as_PID        FROM SAR_Assets
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Maneuver_to_rescue_PID      FROM SAR_Assets,
            $b: Remain_present_for_rescue   FROM PID
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Provide_updates_to_OSC  FROM SAR_Assets,
            $b: Receive_updates_for_OSC FROM On_Scene_Commander
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Notify_OSC_of_PID_location_and_situation    FROM SAR_Assets,
            $b: Receive_PID_location_and_situation          FROM On_Scene_Commander
                DO ADD $a PRECEDES $b; OD;
COORDINATE  $a: Relay_survivor_location_and_situation   FROM On_Scene_Commander,
            $b: Receive_SITREP_update                   FROM Command_and_Control
                DO ADD $a PRECEDES $b; OD;
```

Figure 10. Monterey Phoenix SAR DRM Code Lines 65–13

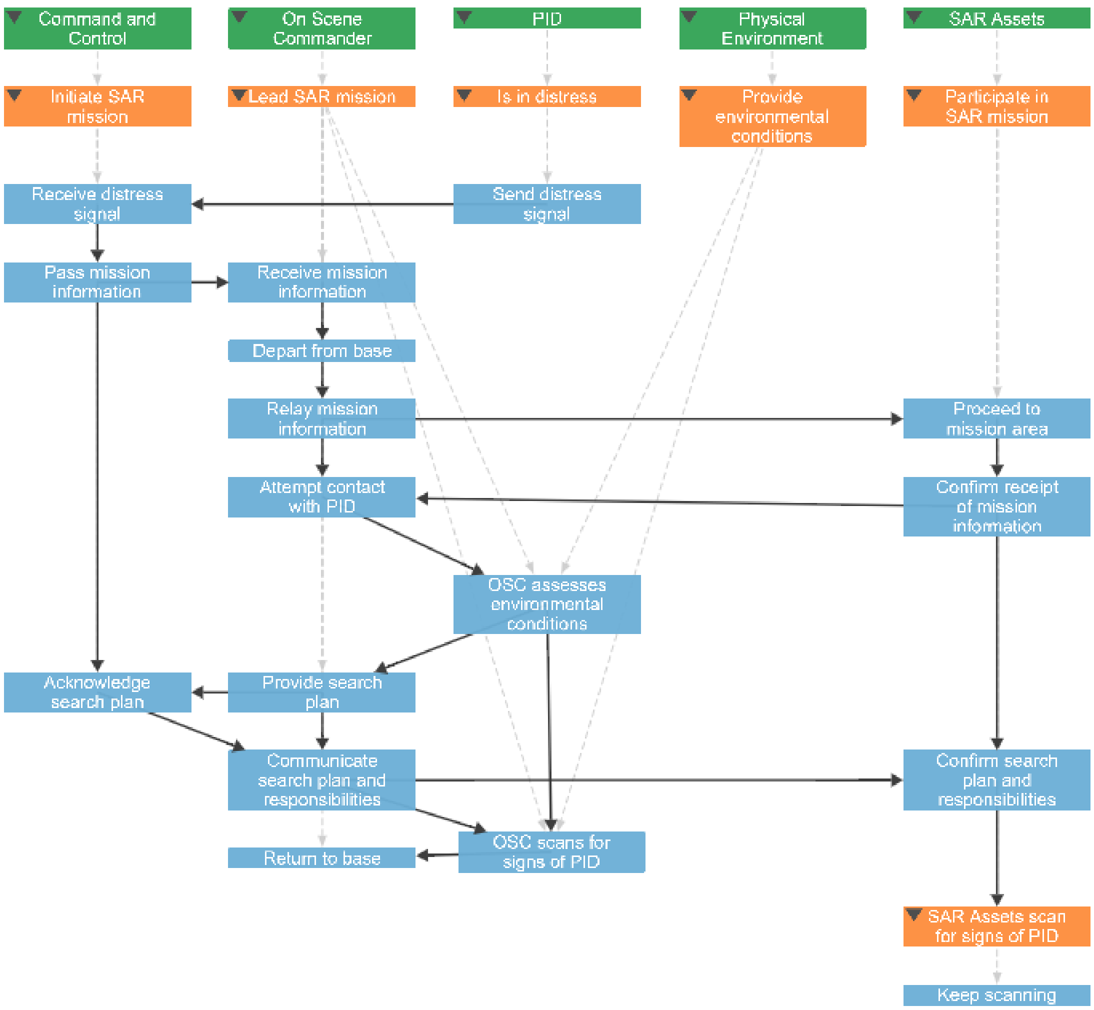

Figure 11. Monterey Phoenix Event Trace: No Object of Interest Spotted

The MP code has been explained, so the automatically generated event traces will now be examined. Starting at the top of Figure 11, note that there are orange boxes underneath each green actor. Since the PID is in distress in this trace, it sends a signal to C2 and activates the SAR system. Now all of the decompositions included underneath the orange composite events in each ROOT are generated. The gray dashed arrows indicate inclusion decompositions (lifelines) under each actor, as well as the share all relationships between multiple actors. Solid black arrows connecting actions indicate the sequence through the various stages of the mission, which are generated via the coordinate and precedence lines of code. Thus, the events of the mission can be traced step-by-step on the black arrows. The Figure 11 event trace concludes with no object of interest provided by the environment, so the SAR assets keep scanning. This lack of an object of interest occurs because in line 40–43, the provision of an object and its relation to the SAR mission is sandwiched by asterisks. This tells MP to accomplish the action zero or more times as opposed one or more times as with the plus signs. Of note, the OSC returning to base at the end of its lifeline will not necessarily occur while the mission is still going on, unless it has some malfunction or fuel issue forcing it back. Therefore, Figure 11 has exposed a portion of the code that is incomplete for the OSC. As such, a specific trigger for the return is needed to make the model more accurate for future refinement.

Figure 12 below is an event trace where the environment provides an object of interest. Note that there now are several more actions generated because of the object of interest's presence. In this trace, it becomes clear why the separate nomenclature was necessary between the environment indicating the object as wreckage and the SAR assets identifying the object as wreckage after assessment. The actions are separate and must be named appropriately to indicate the proper sequence and avoid confusion between the lifelines. Providing updates to the OSC under SAR Assets was an optional action (bracketed in the code) and the reception of the updates by the OSC was asterisked so that MP will do it zero or more times up to the scope selected at run time. If receiving updates occurs, the SAR Assets will have the provide updates action also occur, due to how the code is set up for the COORDINATE interaction. MP generated three more traces similar to Figure 12. One looks exactly like Figure 12 save for the two wreckage blocks indicating the object of interest is not related to the SAR mission. The other two traces omit the optional updates to the OSC while having either wreckage or an object not related to the mission.

The last MP event trace is shown in Figure 13. This trace finally shows the object of interest identifying itself as the PID so that it can be rescued. Since identifying the object as the PID contains an inclusion list, MP generates all the subsequent actions underneath ID object as PID. Figure 13 shows the optional updates to the OSC in addition to the notifications of the PID location and situation. The trace ends with a final relay of survivor information to C2, along with the reception of the information, followed by the OSC returning to base.

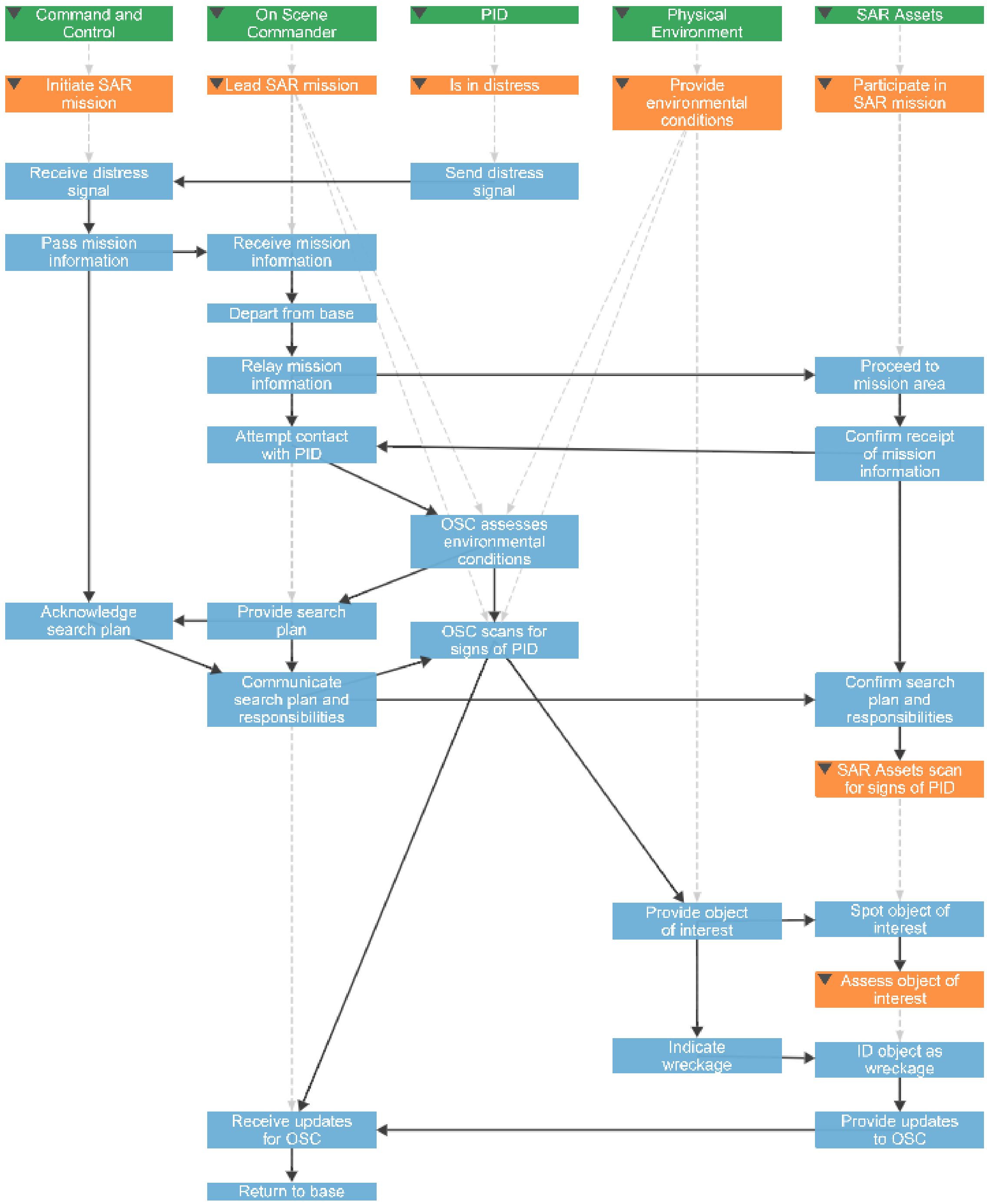

Figure 12. Monterey Phoenix Event Trace: Wreckage Spotted OSC Updates

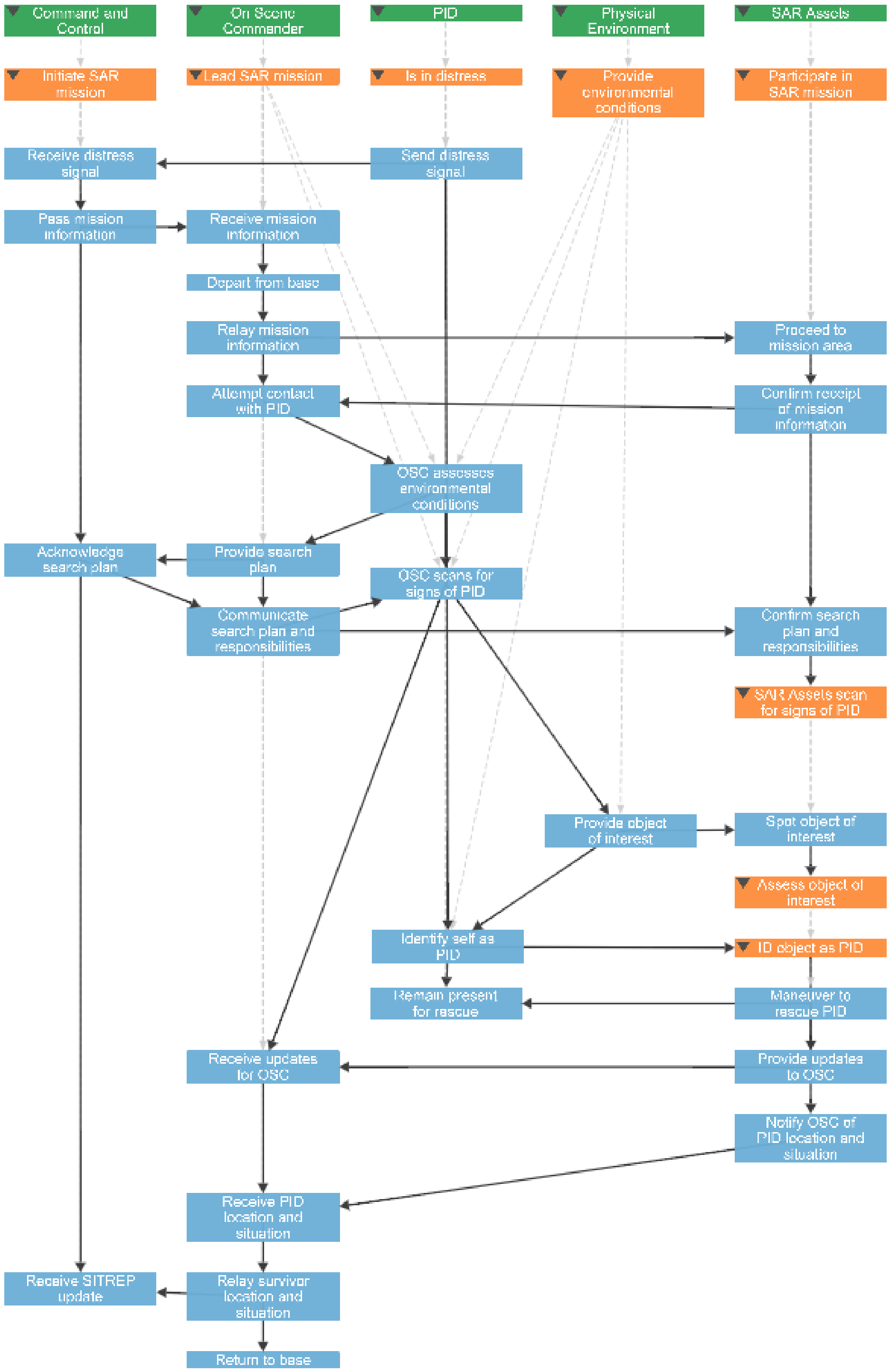

Figure 13. Monterey Phoenix Event Trace: PID Rescued With OSC Updates

2. Model Assessment

In the previous section on sequence diagrams, one of the limitations discussed was the amount of time required to generate multiple sequences for different stimuli within the mission. Obviously these varied scenarios are necessary to evaluate a system's architecture and its interactions with itself and its environment. In this domain, MP has shown that it is a powerful tool for automatically generating such event traces. In all, MP generated eight traces for the code lines presented in Figure 8 and Figure 10. These traces were run on scope 1 in the software, meaning that MP ran looping events (asterisked events decomposed under the root actors) zero and one times. Scope 2 runs looping events zero, one and two times. Scoping options go all the way to five, meaning that each increase in scope can have exponential effects on scenario outputs, especially if the system is complex and contains a large number of looping events. This level of automatic generation is simply not feasible when diagrams are constructed one at a time. Obviously with so many automatically generated scenarios, some will contain no surprises and will look almost like duplicates. This was certainly the case for a few of the SAR DRM scenarios that simply differed by an optional action inclusion or a classification of an object of interest. Inevitably, though, there will be scenarios that show unpredicted or unintended behaviors dormant in the design that cannot be anticipated without extensive modeling. This is where model-based systems engineering techniques such as MP are most helpful. They enable early exposure and refinement of design decisions so that modifications can occur before time and money are wasted further on the process. The longer latent issues in system design go unrecognized, the more costly they become to address.

Another strength of MP when it comes to automatic scenario generation is the ease of refining the models by simply changing code lines. Looking back at Figure 12, suppose an exploration was desired into what occurs with the SAR assets and OSC after an object of interest is identified as wreckage. In a real-world scenario, this would certainly alter the search plan because with identified wreckage, the OSC has an updated position to use as a datum for a new search pattern. To put this into MP code, ID_object_as_wreckage on line 59 would require an inclusion relationship where follow-on actions are outlined. The SAR assets could then request an updated search pattern from the OSC, initiate a new pattern on their own, stay on-scene to explore the wreckage further, or any number of other behaviors. This new set of behaviors could also be linked to the OSC where extra actions could be modeled based upon the responses from the SAR assets. As another example, suppose a modeler was interested in expanding upon the conditions forcing the OSC to return to base before the mission concluded. Optional events could easily include malfunctions and low fuel states that drive the OSC back to base, forcing C2 to launch a replacement asset and perform all the coordination required to establish a new OSC on scene.

These examples are but a few of the ways that the basic code provided in Figure 8 and Figure 10 could be refined and expanded beyond what is presented in this study. A great number of scenarios are possible across a wide range of analysis paths depending on what aspect of the mission or architecture a modeler wishes to focus. For instance, the SAR DRM for this study was written generically to apply to multiple situations, but that does not mean that a specific OPSIT could not translate into MP. The actions and actors would have different names unique to a particular OPSIT, but multiple scenario paths could easily be modeled with looping events in order to study the behavior of the assets in the system and in the environment. The code could also be adjusted to analyze the behavior of a particular type of asset operating within the architecture. If a modeler was interested in how an unmanned vehicle would perform in the role of OSC or as one of the SAR assets, actions specific to that unique asset could be programmed into the code to evaluate its performance and interactions within the traces. Even the environment could be refined for different weather and mission space conditions that would drive different interactions and decisions from the actors. Ultimately, system designs are growing more complex and designers require the means to perform comprehensive and correct analysis in order to make critical design decisions. The sheer number of interactions and scenario possibilities inherent with complex systems can quickly exceed any human ability to generate and predict without automation. In order to stay ahead of the problem, flexible modeling techniques like MP are essential tools for analyzing the intricate system relationships so the best design decisions are realized.

3. Insights

Since the MP models were constructed right after the sequence diagram, it was interesting to see the level of flexibility afforded by the coding. For instance, it was helpful to portray multiple actor inclusions simultaneously when more than one actor was involved in a specific activity like assessing the environmental conditions. Viewing an event trace this way allows the modeler to widen the aperture of abstraction to portray multiple sequences as they might occur in the real world. A sequence diagram might be easier to trace and simpler to look at, but the rigid interactions between assets does not capture all of the intricacies and complexities happening all at once in the mission space. This is important because viewing the entirety of system interactions in a particular trace enhances understanding and ensures that details are not missed. Coupled with the power of automatic trace generation, MP provides a means of predicting and identifying hidden traits so that if undesired behaviors are identified, they can be mitigated or eliminated. A great example of this in the SAR DRM was dealing with the object of interest and how to code it into the program. The figures presented in this section represent the fifth refinement of the SAR DRM code because every attempt to make the object of interest its own actor ended with undesirable and confusing model behavior. For example, in several event traces MP would show the object of interest as "not detected" even if there was wreckage or if the PID was interacting with the SAR Assets. The desired behavior was for an object of interest to be discovered in the physical environment lifeline and then once examined, classified. When the object was its own actor, this was a difficult result to achieve. Even when the object did behave as desired, its lifeline only decomposed the single action of whether or not it was related to the SAR mission. Part of the reason these issues arose was due to the input code, but in the process of refining the model it became apparent that perhaps the object needed to be modeled differently across all of the techniques. This was an important insight that eventually affected every model in the study.

A similar insight occurred when attempting to model recurring SITREP updates among the OSC, SAR Assets and C2. Like the object of interest, this was a situation where the code was producing undesired and confusing results in the event traces despite several attempts to achieve the correct behavior. In traces where the recurring SITREP interactions got close to the desired behavior, the diagrams were overly crowded with arrows and boxes, making the trace difficult to view and understand. The general rules of the mission narrative exist to simplify the models by means of assumed behaviors that need not be depicted. Since the SITREPs could be assumed as a matter of course throughout the layers of communication occurring during the mission, and were unnecessarily causing problems with the models, the solution was to include them in the general rules. This was another important insight gained from MP that affected all of the models in the study.

While it does take some time to understand some of the grammar and language rules with the MP code, there is no denying the power of MP as a modeling tool. Even with a generic SAR scenario, MP provided a wealth of insights for how to model the mission in its own code, and gave insights into modeling across the rest of the study. These kinds of insights are most helpful to designers because intricate relationships are difficult to see when complex systems have an overwhelming number of moving pieces. If unintended behaviors exist in a system, they need to be maximized if they are desirable and minimized or eliminated if they are undesirable. This must occur as early as possible to save time and cost, and techniques such as MP help eliminate unexpected and undesired behaviors from a system's architecture.

E. SPIDER DIAGRAMS

In model-based systems engineering, spider diagrams are primarily a brainstorming and planning tool because their structure naturally allows for stimulating ideas. Similar to mind maps and concept diagrams, spider diagrams use components of each and apply them to a unique graphical representation. From a mind map, the spider diagram utilizes a radial construct whereby a central idea, function, or component is placed at the center and relationship connections represented by lines or arrows branch out to other functions, actions, or components. Whereas a mind map tends to focus on only one central idea, a spider diagram is more structurally flexible because there is no set of rules for how the diagram must look or what it must contain. This construct idea is more like the concept diagram where a single representation could include functions, actions, components, or whatever piece of a system is desired. The key is labeling the relationship connecting lines so the diagram is not confusing no matter how varied the ideas become as they branch out from the center.

In terms of construction, the connected components of a spider diagram are usually placed inside of a shape for organization and to give the line connectors something to attach to other than just words. From the central component or group of components, it is then simply a matter of "fleshing out" relationships to the rest of the components or functions and labeling the relationship arrows appropriately so that anyone viewing the diagram knows how one component relates to another. For ease of readability, different colors and shapes could be used to denote different interactions or relationships, but this is not a requirement. Additionally, for complex systems, consideration should be given to using abstraction on higher levels or breaking up subsystems into multiple diagrams. This way the diagram stays organized so it is easy to view and understand. Too many arrows and components will quickly clutter a spider diagram and defeat the purpose of constructing it in the first place.

Although spider diagrams are useful for the creative process, they should not be overlooked as a tool in more complex modeling efforts. Regardless of how concrete a system's functions seem, there is still benefit in constructing spider diagrams so that a multitude of relationships can be viewed in a single diagram. This is a helpful visualization that promotes communication between different designers within a project, and is also helpful for communicating relationships to the customer.

1. Model

Figure 14 is a good example of a high-level spider diagram for the overarching SAR Activity Context from the DRM. Innoslate can construct these diagrams automatically based on relationships that exist from IDEF0 models, hierarchy diagrams, activity diagrams, and use cases. They also can be built independently based on the needs of the modeler. Innoslate allows the user to include all possible relationships that exist in a system from one of these other modeling techniques, and it also gives the user the ability to simply look at a traceability representation. This is the nature of Figure 14. A third option exists to customize the diagram so that the user can select what component blocks are shown along with the relationships. All three of these options are useful depending on what relationships a modeler wishes to portray in the diagram. Showing every possible relationship can generate an extremely cluttered diagram, however, so caution should be used when diagramming all relationships for a complex system.

Figure 14 acts as a high-level concept diagram for how the modeling of the SAR DRM activity context has evolved. Unlike the OV-1 from Chapter III that was generic in nature, the spider diagram portrays some specific detail on decompositions and performances. Since the SAR activity context is the main focus of Figure 14, it is in the middle of the "web." All operational activities (OA nodes) that decompose the SAR activity context form the first set of branches out from the center. What is interesting about the spider diagram is that the physical context is also present along with a few entities that decompose it. In a hierarchical diagram, functions must decompose functions and components must decompose components, but in the spider diagram, there is flexibility to show both simultaneously along with their branches. As mentioned previously, the use of color and line labels are not necessarily required for a spider diagram, but when a view contains a mix of functions, physical components, decomposition relationships and performance relationships, they become essential for understanding the diagram. Of note, depending on which component or function is placed at the center of a "web," the diagram can look slightly different even if it appears that all the same nodes are present. For example, Figure 14 does not branch completely out when it comes to relationships off the SAR physical context. It does show the five major physical assets but it stops short of depicting the physical subsystems of the OSC because that is beyond the scope of this particular spider diagram for the activity context.

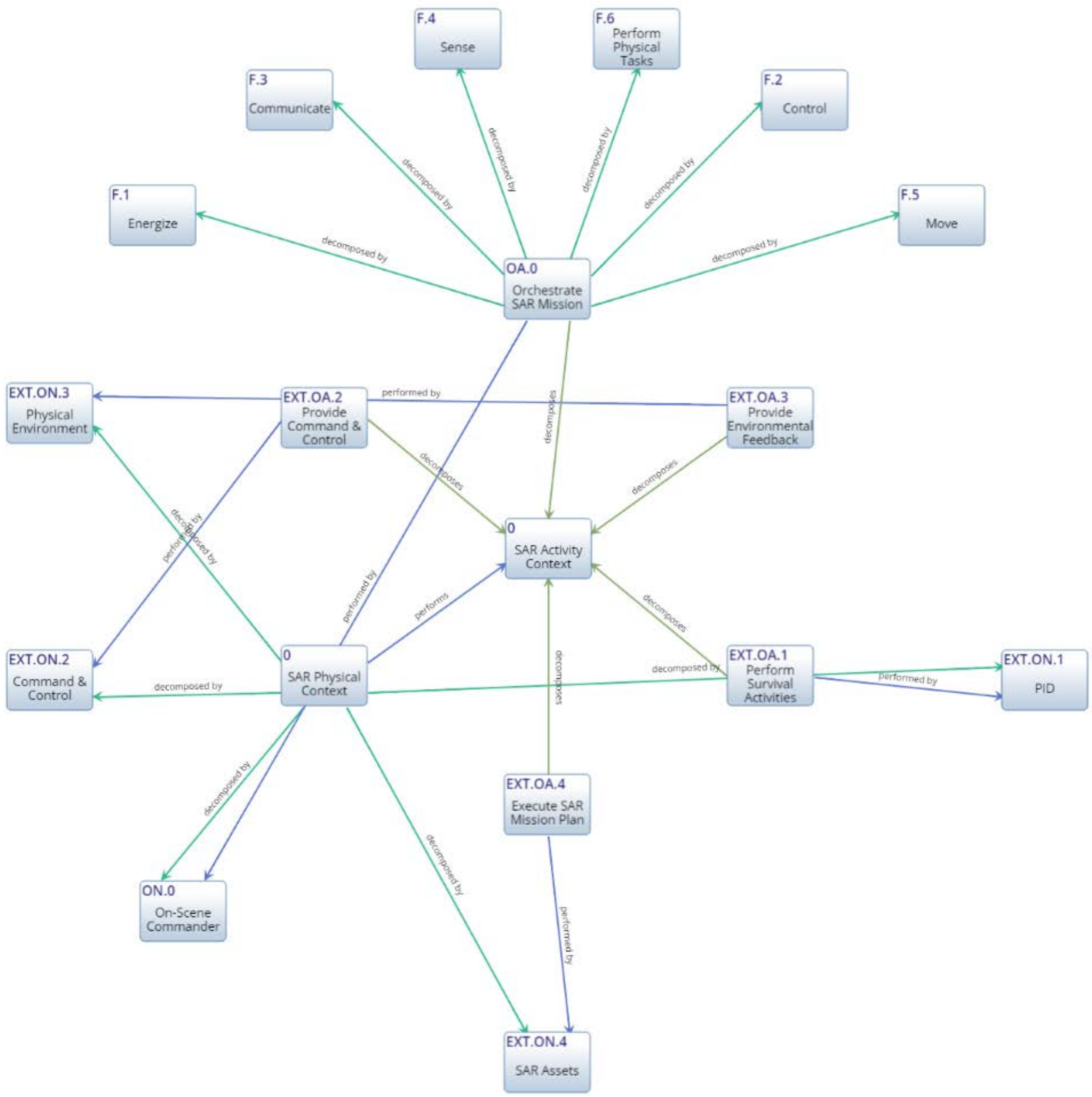

Figure 14. SAR Activity Context Traceability Spider Diagram

A spider diagram with the physical context at the center of the "web" would decompose all physical components, but by default omits the furthest decompositions for the activity functions. This is mentioned to show how Innoslate generates spider diagrams and to illustrate that while spider diagrams are flexible, there should be a limit to how far a branch extends, which is wholly dependent on what is in the center of the "web." Otherwise, the scope of the diagram is too wide and the diagram can lose its contextual correctness and become confusing.

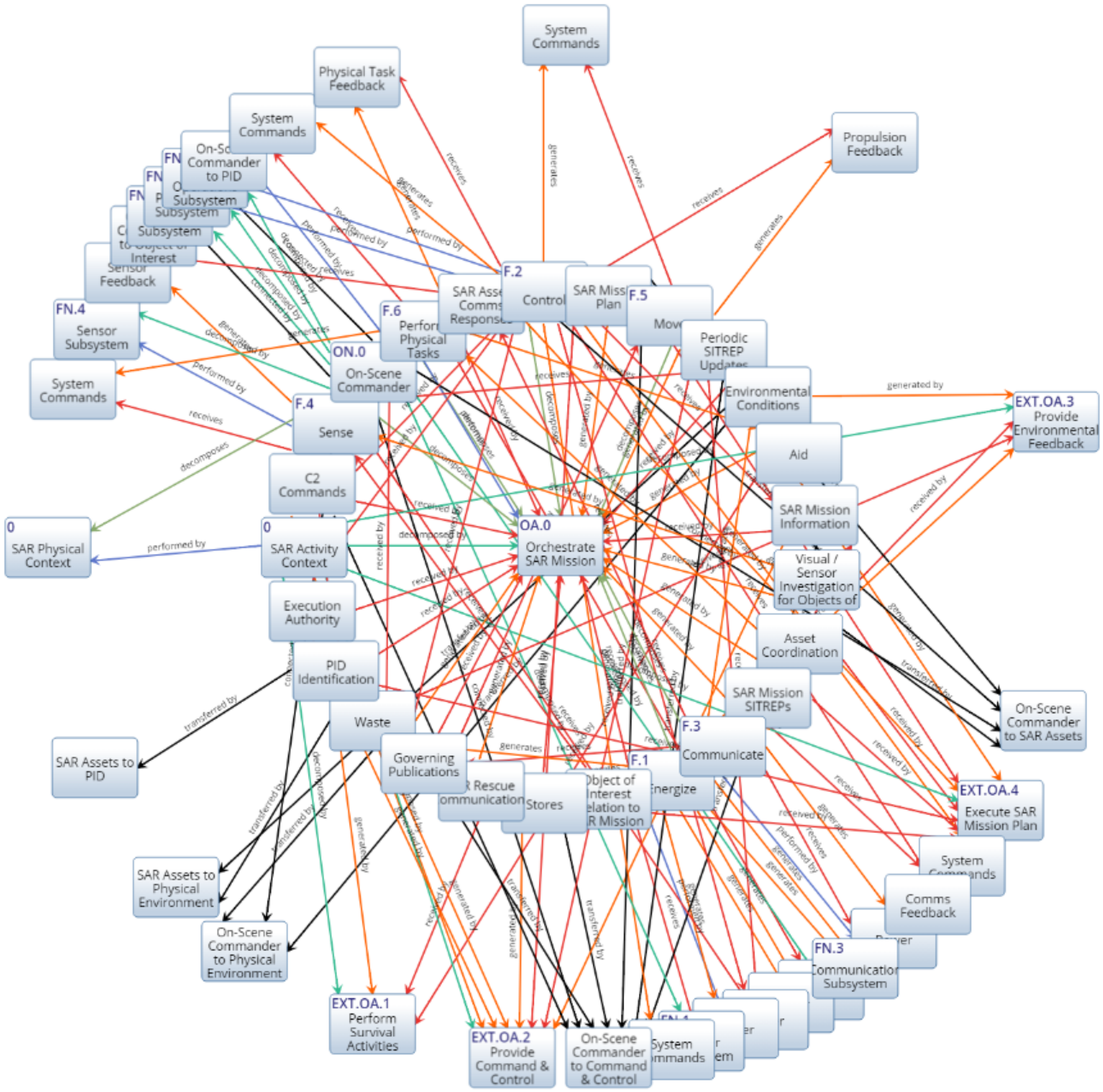

Figure 15. OA.0 Orchestrate SAR Mission All Relationships Spider Diagram

Figure 15 illustrates how too many depicted relationships for a complex function or component in a system can render a spider diagram useless. Figure 15 is undecipherable and there is no way to ascertain anything from such a diagram. A person constructing such a diagram by hand would quickly realize that it is too complicated, but when such a diagram is generated automatically using software, such pitfalls are often an undesirable result. This is why Innoslate allows a user to customize and simplify a spider diagram. Perhaps at a deeply decomposed level, it might be feasible to diagram every possible relationship for a particular component or function, but on the higher levels, there are too many relationships to appropriately display in a two-dimensional view.

Figure 14 and Figure 15 do not represent the totality of spider diagrams that are possible with the data in Innoslate for the SAR DRM. The reason for not considering all of them is that at some point, they become redundant. Many views are related and the only difference is what function or asset appears in the center of the "web." Constructing multiple spider diagrams for a system, even if they are closely related, can be useful in showing relationships to other designers and to the customer, but a small sampling suffices for this study.

2. Model Assessment

The functions and components in the spider diagrams have been addressed previously in the hierarchy-type diagrams and IDEF0, so the question arises as to the usefulness of the spider diagram. The spider diagram's usefulness depends on the purpose of the modeling effort. If the goal is to brainstorm relationships between functions and components at the start of a project, then the spider diagram's flexibility is useful for stimulating creativity. If the goal is to create a simple diagram for communicating expectations and relationships to customers or other designers who may lack a heavy background in model-based systems engineering, then the spider diagram also is a user-friendly graphical representation. When a spider diagram is constructed based on a detailed IDEF0 already in existence, however, its design may have limited usefulness to a modeler since so much detail is already represented in the IDEF0. This does not invalidate the spider diagram in any way, but it does highlight the fact that the many modeling techniques and languages available for the systems engineer may create overlaps such that extra efforts to construct additional models, such as a spider diagram, may not be needed. If another model provides an adequate representation of what the modeler is pursuing, then it is important to realize when generating models so closely related is a waste of time. However, since one never knows how a particular individual will respond to or understand any model representation, software such as Innoslate that can generate spider diagrams based on information already input into a database makes it easy to build extra models for any purpose even if they provide redundant information.

3. Insights

One of the most useful attributes of the spider diagram is its flexibility. Figure 14 has shown how easily functions and components can be combined in one diagram along with their respective relationships because of this flexibility. On the other hand, Figure 15 has shown that flexibility without boundaries can be a problem so it is important to remember to properly scope a diagram. In terms of the SAR DRM, it is helpful to have a view where functions and physical components can exist on the same diagram with simple lines and descriptors showing relationships. While not as detailed as an IDEF0, the spider diagram affords the opportunity to see everything on one page and this can make the overall view of a system easier to comprehend. Furthermore, when using Innoslate, it is sometimes difficult to remember what physical components perform what functions and how different actions and assets are connected. Usually, the modeler must open individual entities in Innoslate to ascertain the connections, but with a spider diagram, they are all portrayed in an easy-to-view graphic. If more detail is desired, a simple opening of a relationship entity will provide every detailed interaction existing in the database.

Like the hierarchical diagrams, spider diagrams will continue to be useful in the SAR DRM as further decompositions delve into areas like specific units to use for the OSC and SAR Assets. As a brainstorming tool, the spider diagram easily lends itself to creative efforts in seeking new relationships between entities and highlights how their interactions could be improved or redesigned in order to make the overall SAR architecture function more cohesively. The spider diagram is an excellent way of staying organized during such an effort and provides a solid means of tracing the evolution of ideas as new concepts are explored.

F. ACTIVITY MODELS

Activity models are executable system representations that can be simulated in order to observe how components interact over time. Interacting components within an activity model may be multiple systems interacting with the environment, or they could be various subsystems performing within a system. Like many of the models already presented in this study, the best place to start with an activity model is the high-level context of a system interacting with external systems and the environment. After these context relationships are established, individual systems and subsystems of interest may be decomposed further to achieve simulations for components on the lower levels. The point of these executable simulations is computing parameters such as mission completion time and asset resource consumption so that different system configurations and architectures can be objectively assessed with performance data.

An activity model works by placing activities on a timeline so that the simulator can show duration and resource consumption throughout the course of a system's interactions. The interactions could be the system completing a mission or a particular task either internally or externally. Activity models can also help describe the flow of control within a system as it pertains to something like complex business operations or use cases in a business process. Figure 16 below is an example of an activity diagram for a piece of software that creates customer shipments. It will serve as an illustration of some of the capabilities and components of an activity model to aid in understanding the more complex diagram for the SAR DRM.

Figure 16 begins with the overarching activity of Create Shipment. By definition, activities in an activity diagram "specify the coordination of executions of subordinate behaviors using a control and data flow model" (Visual Paradigm 2015). This means that any system can be modeled as a network of activities and inside of each activity are the subordinate processes and data flow required to complete the activity. This is the context of Figure 16 because Create Shipment is the activity box, labeled in the upper left of the diagram, and everything inside the box is the flow required to achieve the activity. Moving upward from a single activity is analogous to looking at a higher context for the system, so in the case of Figure 16, creating the shipment may be one of many activities the software could perform. Each activity would have a corresponding decomposition like the one presented in Figure 16, but the activity would not be portrayed on a higher-level diagram for simplicity.

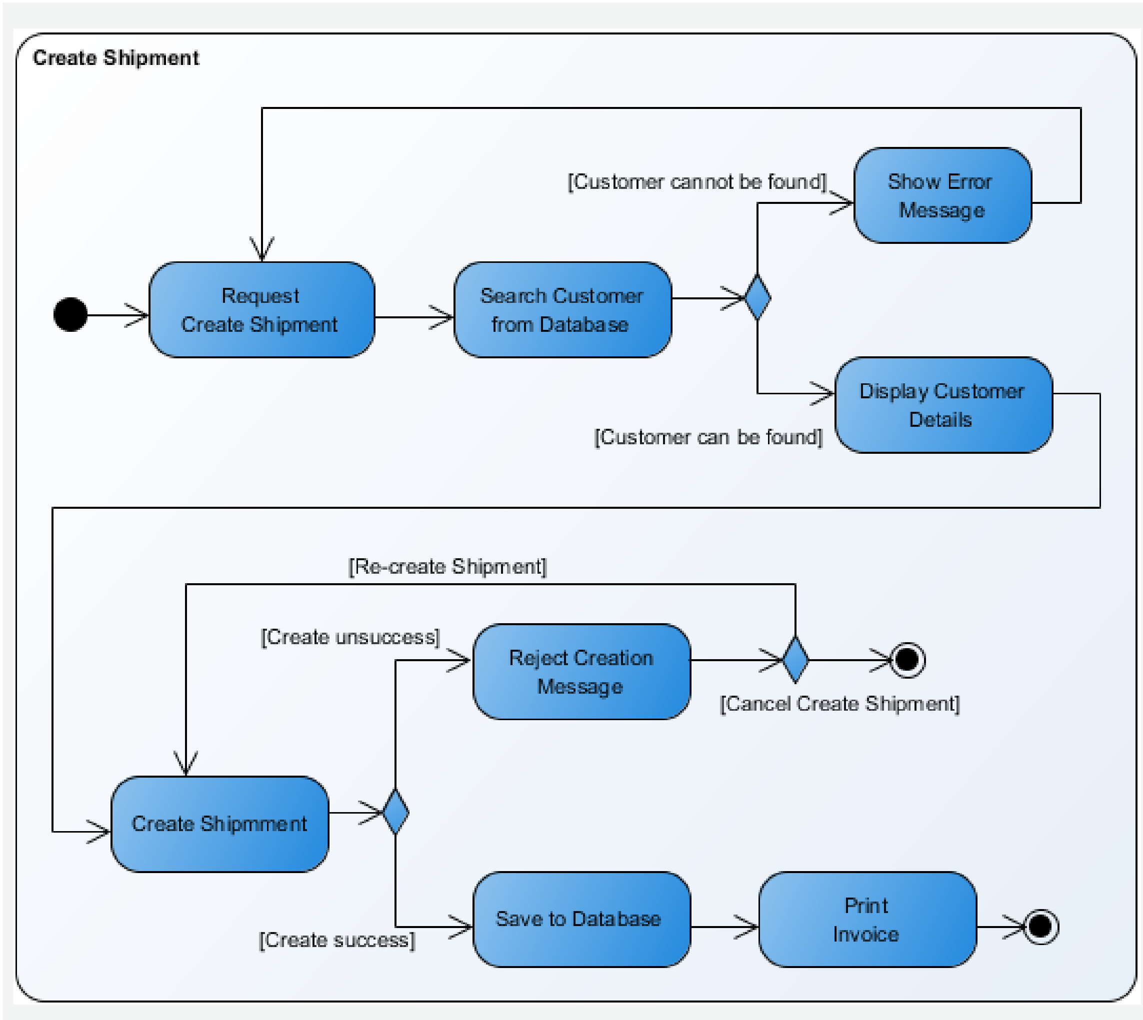

Figure 16. Example Activity Diagram (from Visual Paradigm 2015)

On the inside of the activity box, the black circles are initial and final nodes. The initial node is the single black circle while the final nodes have the outer rings. Initial nodes start the flow when their parent activity is invoked and final nodes stop all flows in the activity for a particular branch. An activity may have more than one initial and final node within it depending on where flows start and stop within the activity. For instance, Figure 16 has an initial node that activates when a shipment request triggers the Create Shipment activity. On the lower right, two final nodes end the flow either with a canceled shipment or a successful shipment with a printed invoice.

The dark blue boxes are actions within the activity that are not decomposed any further. In essence, an activity is a behavior composed of individual elements, and those elements are the actions. “An action may have sets of incoming and outgoing activities that specify control and data flow to other nodes, but an action will not execute until all of its input conditions are satisfied” (Visual Paradigm 2015). The lines connecting the actions represent the control flow itself, and they may be labeled as necessary in order to ease interpretation. The blue diamonds represent decision nodes where the flow can continue on one path or another depending on the input received at that decision node.

Knowing the notation and definition of the symbols makes navigating through Figure 16 straightforward. After the request is made to create a shipment, the customer database is searched. If the customer cannot be found, the decision node forces the flow toward an error message that then routes the flow back to the initial request to try again. If the customer is found, the customer’s details are displayed and the shipment is created. If the shipment is created successfully at the next decision node, it is saved and an invoice is printed as the flow moves to the final node and is stopped. If the shipment creation is unsuccessful, a rejection message is displayed and a different decision node can cancel the shipment entirely and end the flow, or it can return the flow back to the creation action in an attempt to re-create the shipment.

In order to simulate the activity in Figure 16, duration value ranges are input into the simulator as well as probabilities of success at the decision nodes. Then the simulation could be run as many times as needed to ascertain an expected duration and probability of success for the entire activity. While not depicted in Figure 16, the possibility exists to assign resource consumption to individual activities and actions. For instance, printing the invoice obviously takes paper and ink supplies, so estimates for those consumables could be input into the simulation in order to predict needed supplies over a period of time for a certain range of successful shipment creations. It is easy to see how powerful and practical these types of executable models are when it comes to complex systems. Used on systems already in existence, accurate executions can be used to predict needs, or they can be used to target potential activities and actions that need redesign in order to boost performance. For systems that have not been designed yet, these models provide a means to study desired performance characteristics as a road map to achieving them in prototyping and final design.

1. Model

Figure 17 presents an activity model for the SAR DRM. In Innoslate, this particular view is called an action diagram. The look is slightly different from the one provided in Figure 16, but many of the same components—action boxes, control flow arrows, start nodes, and end nodes—are still present. The major difference in Figure 17 is each major asset is given its own "swim lane" that is analogous to the event trace "lifelines" that have been presented in previous models. The linear view is one of a few ways an activity model may be portrayed. When event traces already exist, this view is a seamless transition to the activity model because it shares a similar structure.

In Figure 17, each horizontal line is labeled with the asset owning the lane, and all of the events on the lines are logically concurrent. This means the first activity on each lane will execute simultaneously with all of the others on their own lines. It is obviously undesirable for all of the M.A. actions to execute at the same time, so constraints are necessary to ensure that the actions occur in the proper sequence. This is usually accomplished in simulations via triggers, which are special inputs that prompt an activity to execute. This is the function of the green parallelograms in the diagram. Since the activities will not execute until their specific trigger arrives, all triggers must be received at the appropriate destination activity in order for the activities in the simulation to properly compute (Giammarco, Hunt, and Whitcomb 2015). For example, M.A. 1.4 is the OSC relaying mission information to the SAR Assets. Mission information is the trigger to begin M.A. 1.5 where the SAR Assets proceed to the mission area.

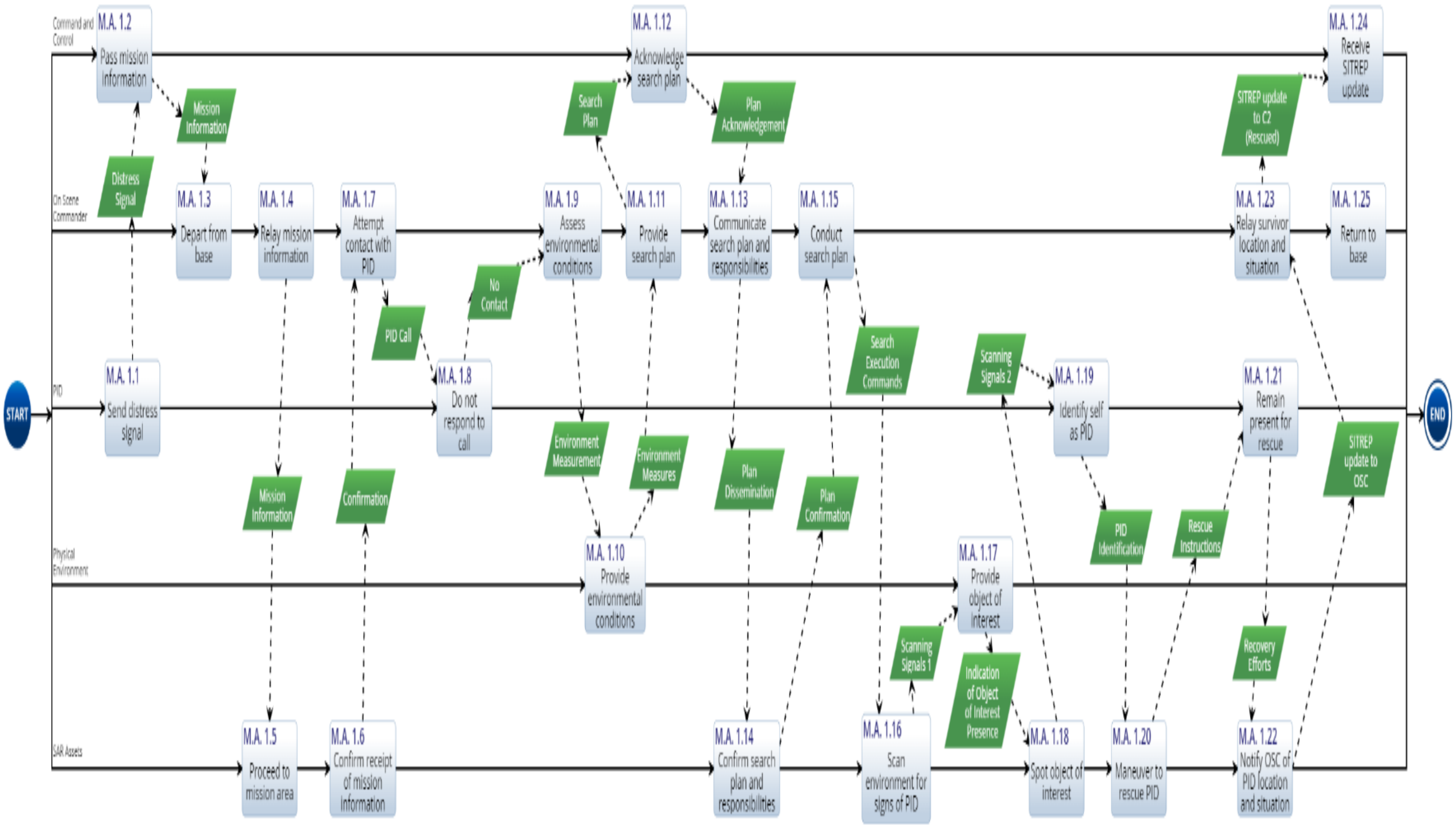

Figure 17. SAR DRM Action Diagram

Actions M.A. 1.5 and M.A. 1.6 occur in sequence, with or without a trigger from M.A. 1.5 to M.A. 1.6, because they are on the same branch. Were it not for the confirmation trigger going back up to the OSC line in M.A. 1.7, that action immediately would have followed M.A. 1.4, executing simultaneously with M.A. 1.5.

Sometimes it is desirable for actions to occur simultaneously. The way to accomplish this is by omitting the trigger that forces actions on different branches to occur in sequence. An example of this occurs at the end of Figure 17 on the OSC line. There is a trigger from M.A. 1.23 where the OSC gives a SITREP update to C2 about the PID rescue and M.A. 1.24 has C2 receiving that SITREP. There is no trigger for M.A. 1.25, however, meaning that it will execute as soon as M.A. 1.23 is complete because both actions are on the same line. This is intended because it shows that the OSC can communicate with C2 while returning to base and it is logical to model it this way because an OSC would not stay on scene needlessly once the mission is complete. This example, coupled with the one presented in the paragraph above, illustrates why it is important to make sure triggers are positioned appropriately. If they are not, the control flow will produce erroneous data because actions will not execute in the proper sequence. Finally, because triggers are often used to specify information that is passed along with control, they typically have a noun-oriented name, which contrasts with the verb naming convention used for the actions in the model (Giammarco, Hunt, and Whitcomb 2015).

Like the example activity model presented in the background section, once the symbols and notations are understood, it is simple to navigate through Figure 17 by tracing the control flow arrows through all of the actions. Many of the actions will be quite familiar at this point because of naming continuity throughout previous models in this study. Note that there are a few added actions that must be present to add accuracy and realism to the simulation. An example of this is M.A. 1.3 where the OSC departs from base after receiving the mission from C2. The OSC proceeding to the scene is a real action that takes time and thus it must be accounted for in a simulation attempting to ascertain a total mission time. The same holds true for M.A. 1.25 where the OSC returns to base.

Once the model is constructed and checked for control flow, the next step is assigning duration values to the actions. In most simulators, this occurs by assigning a probability distribution to an action that will generate an output time for a given simulation run. If there is a set duration for a particular action, a single value may also be assigned so that the simulator considers that exact value in every simulation. The choice of probability distribution is usually based upon the action it is assigned to, but is ultimately the determination of the individual performing the simulation. Each distribution will require a set of values to define it, another decision that is up to the modeler based on previous design data or observed operational performance. The actions in Figure 17 contain a mix of normal and exponential distributions for simulation purposes in order to produce reasonable hypothetical results. Of note, the triggers may have values or distributions assigned to them as well, if that is appropriate for a particular model.

Before any detailed analysis is explored for an activity model, it should be checked for continuity and correctness. Since the model will execute based upon its setup and input values, any logic problems or continuity gaps will quickly arise when the simulator is run. This is an important step in debugging the model and verifying that the parameters set for individual actions are reasonable and accurate. Figure 18 shows the results in a Gantt timeline chart from running the simulator on the SAR DRM activity model. Moving from left to right, the first column shows the action’s title and the second column gives duration as a result of the input probability distribution or a set value. The right side of the diagram is a graphical representation of the action executions in order.

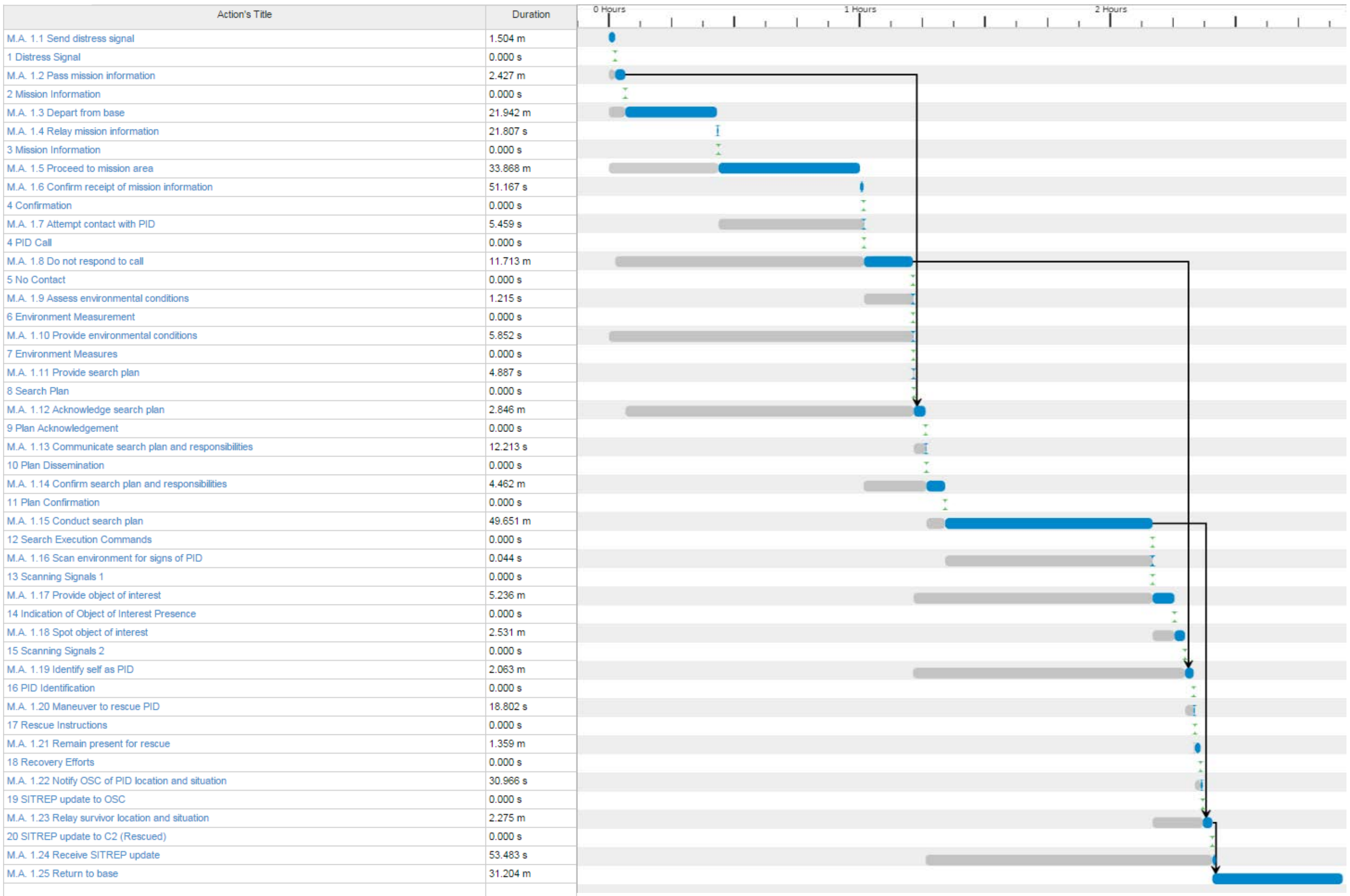

Figure 18. SAR DRM Activity Simulation Gantt Chart

In terms of verification, a simulation result like Figure 18 allows the modeler to see if all the actions are occurring in the expected logical order and if there are any deadlock conditions preventing action execution. These deadlocks are caused by unsatisfied conditions in the in the model, like an action awaiting a trigger that has not been sent. Figure 18 captures all 25 actions from the action diagram along with all 20 triggers and they all occur in the proper order on the column list and on the Gantt timeline. Were this not the case, an inspection of the model for the out-of-sequence actions would be necessary to ensure that the triggers are placed appropriately and that the actions are properly ordered on the individual lanes. If the model is out of order, the triggers can be toggled or actions can be renumbered in order to correct the logic of the simulation. On the Gantt chart side, the actions and triggers will be in order so long as they are in the desired sequence on the left side of the diagram. The blue bars indicate actions that are executing and their length is determined by the time value from the simulation. The gray bars are enabled actions awaiting a trigger before they can execute. The black arrows are precedence relations imposed by the order of the actions on the branches in the model. For instance, M.A. 1.2 has C2 passing the mission information to the OSC and there is a long sequence of actions occurring with sequenced triggers for all the other actors before C2 acts again in M.A. 1.12 to acknowledge the search plan. M.A. 1.12 is enabled as soon as M.A. 1.2 is completed because they are on the same branch line and that is the meaning of the gray bar preceding the blue bar action when the search plan trigger finally executes M.A. 1.12. The arrow is a means of indicating this precedence for traceability at a glance when viewing the timeline.

In terms of validation, the simulation output allows the modeler to ensure that all of the correct activities are present and that the model makes sense from an operational perspective, not just simply from a logic standpoint for having the simulation execute without error. The simulation output is also an excellent way to ensure that the numbers generated are accurate representations of the real world. In Figure 18, most of the durations supplied by the simulation make sense within the parameters of the probability distributions that were defined for each action. Some of the actions have somewhat unrealistic durations, however, and should be adjusted before proceeding with any in-depth analysis using the model. An example of this is M.A. 1.4 where the OSC relays mission information to the SAR Assets. It took C2 just over two minutes to pass the initial mission information to the OSC so it is reasonable to assume that it will take about that amount of time to relay the mission information to the SAR Assets. The simulator has M.A. 1.4 taking only about 22 seconds, which is unrealistic. M.A. 1.9, M.A. 1.11, and M.A. 1.13 are also examples of actions whose durations should be adjusted because it will certainly take longer than a matter of seconds for the OSC to assess the environmental conditions, provide the search plan to C2, and communicate the search plan to the SAR Assets. These inaccuracies can be addressed by tightening up the parameters of the distribution functions on the model to ensure a narrower scope of numbers that are possible for the simulator. It may still be feasible to run the simulation a few times first, however, to ensure that the outputs are consistently inaccurate before adjusting numerous distributions. In the case of Figure 18, five simulation runs were performed to ensure that the model was producing consistent duration values. Based on those runs, it is apparent that some of the distribution functions should be adjusted to reflect real-world durations more accurately before proceeding with any in-depth activity model analysis.

A final output from the action diagram simulation in Innoslate is the action utilization bar chart that depicts a percent utilization of each action's duration against the total mission time. Innoslate automatically generates these diagrams based upon the order of events in the Gantt chart, but it also allows a modeler to sort the actions for their individual percent values as in Figure 19, making it easier to compare the actions and visually inspect the values for anomalies. Inspecting this view is a means to accomplish some final checks on the simulation to ensure that the output values are reasonable for a real-world situation. Since the total mission time for this simulation was 2 hours, 55 minutes, and 37 seconds, some large percentages on actions like conducting the search plan and the OSC and SAR Assets moving to and from the scene are understandable. Some inaccuracies that were identified in the column view of the Gantt chart are present in Figure 19 as well, further verifying that those actions need some probability distribution adjustment to reflect a more realistic time output in the simulation. Note that the triggers also are accounted for in this view. If they had any duration, they would contribute to the total time just like the actions.

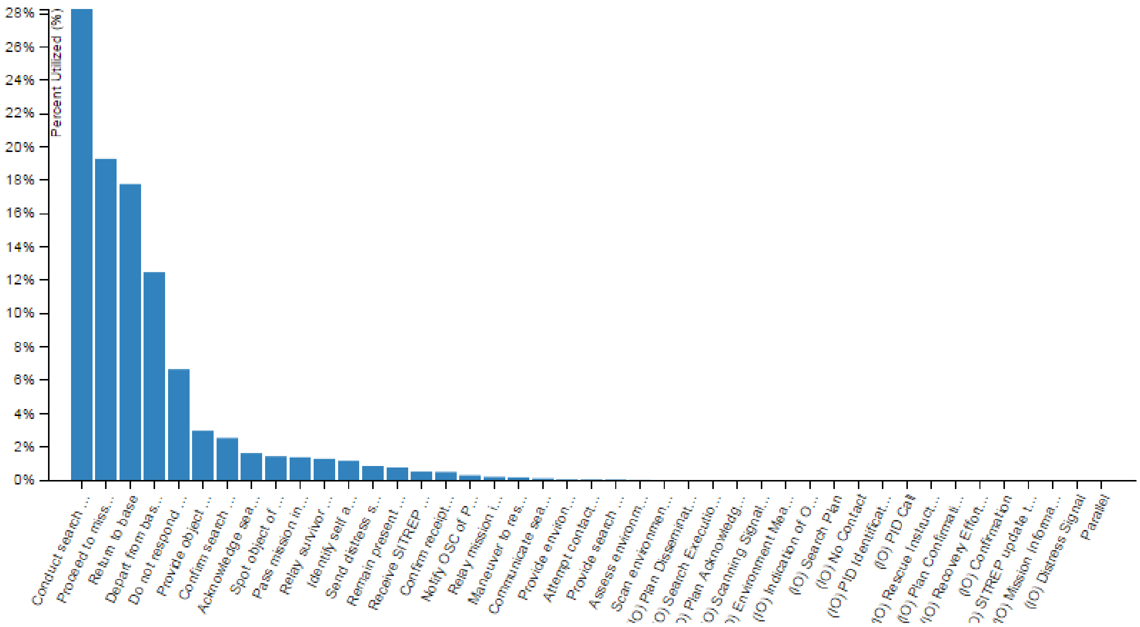

Figure 19. SAR DRM Action Utilization Simulation Output

2. Model Assessment

The major strength of activity models is that they can be executed via simulation. It aids in understanding a system's architecture and interactions to use many different kinds of models to describe a system, but when the modeler can execute those interactions accurately with the inherent complexities in the design, then a great deal of understanding is gained on important behaviors. Since modeling is aimed at understanding system interactions in order to meet requirements and solve design problems as early as possible, activity models provide the means to predict how the system might behave and what resources it might consume. Although no model can capture exactly how a system will perform and operate before it is actually built, this kind of executable modeling is a powerful tool for identifying early behavior issues and areas of the design that need rework because the later these problems are identified, the more costly they become.

Another strength of activity models is their flexibility for incorporating hidden actions that are not expressible in other techniques. Some models like IDEF0 and hierarchy diagrams have a level of abstraction that makes these assumed types of actions difficult to predict. Other models like the sequence diagram simply do not have the flexibility to show a single entity performing an action that does not involve another actor or component in the system. At some point, these actions must be accounted for because they take time to perform and often consume resources. Without an executable simulation like an activity model, these important behaviors may be missed and that kind of oversight in early design development could have far-reaching effects throughout the course of a system's life cycle.

When tackling complex system design with activity models, it is important to remember that not every interaction for the entire system must be in a single model. Figure 17 represents a straightforward, linear sequence of events but as decision points, loops and optional activities are added for more complex sequences, the model can quickly become untenable. This can nullify the usefulness of the diagram and can lead to long simulation times and difficult error detection should any issues arise during execution. This is not to say that activity models should intentionally omit important details and system interactions because that negatively affects the accuracy of the entire modeling effort. Instead, to avoid some of the complexity pitfalls, a modeler should consider layering and breaking up interactions to make the models easier to manipulate. Recall that this can be accomplished via a structure similar to the Figure 16 example. The overall activity box was for creating a shipment and that activity was part of many functions performed by a piece of software. The shipment creation activity box would then be present on the higher level diagram for the overall system, but then a separate model would detail what occurs inside the activity to include all the loops, decision points and optional activities. Figure 17 could be broken up in a similar fashion. Not all of the activity boxes would need a detailed breakdown, but suppose more information was desired on the exact process of the OSC providing the search plan. In this situation, M.A. 1.11 could be broken down into constituent sub-actions in much the same way as Figure 16. There could easily be a number of decision points, loops and optional activities for the OSC depending on what information is available about the environment, mission space and nature of the rescue at hand. This would not necessarily be desirable to annotate on a higher level diagram like Figure 17 and so M.A. 1.11 would get its own simulation. Then, instead of a probability distribution, a narrower range of values could be the input for the M.A. 1.11 box to enhance the overall accuracy of the simulation. This decomposition could occur as many times as needed for any action box depending on what information is desired for the simulation output.

An additional benefit of having multiple activity models versus displaying everything in a complex and comprehensive view is ease of presentation. One of the frustrations with activity models is that as they become more complex, they are increasingly difficult to portray in a single view without distorting the picture. Figure 17 is a good example of this issue. Even with only 25 action boxes, it is difficult to properly crop and align the diagram to fit it to the page. This is an issue regardless of the orientation of a standard page, often making it necessary to cut the diagram up in order to fit it into a slide show presentation or a report. A modeler must be able to communicate these models to the stakeholders and ease of presentation is a factor that cannot be overlooked in the translation process.

3. Insights

It is often presumed that an executable action or activity model must always be a cumulative effort that only occurs toward the advanced conceptualization stages of a system design. While these activity models are often completed toward the end of a modeling effort, this does not have to be the rule. As long as a particular component or process is understood to the degree that modeling it via simulation would be accurate and beneficial, it can be accomplished at any point. In the case of the SAR DRM, it is clear how powerful a simulation tool can be when it comes to evaluating a system architecture designed to accomplish a mission. The format of the action diagram was particularly helpful since the “swim lanes” mirrored the major actors’ lifelines from previously presented models. This made translation simple for understanding and modeling the linear sequence presented in Figure 17. The format also allowed for the addition of some extra actions that were assumed in previous models. For example, the OSC leaving base and returning at the end takes time, so the action diagram gave the opportunity to express such actions and where they occur in the overall timeline. The action diagram also provides flexibility with regard to assigning values or probability distributions to the various actions. There is even an option to perform a Monte Carlo analysis on those distributions when executing a simulation. Although the activity model in this section was not focused on any kind of resource consumption, this too is an option so that the modeler can assess cost alongside time when executing simulations.

Emphasis has been placed on the fact that executable simulations expose unwanted or undesirable behaviors in order to address them as early as possible. To that point, there has been much discussion in this study's other models about the decision to remove the object of interest from the major actors list and ultimately it was the action diagram that brought about this insight. Figure 17 originally had six "swim lanes" much like the original sequence diagram and Monterey Phoenix trace but as the diagram developed, it became clear that the object of interest must be provided by the environment rather than have its own lane. The problem with the object of interest having its own lane is it presupposes that an object will always be present in the mission space. This is not an accurate representation of SAR missions. Furthermore, the actions and triggers for the object were increasingly difficult to integrate when it was its own actor. Action names like "be present in mission space" were needed to link the object to the other actors to continue the flow of the mission. These interactions were awkward at best and made other models difficult to connect with each other as the modeling effort progressed across multiple techniques.

Early simulations and tracing through the sequences of events from Figure 17 proved that the object of interest did not behave properly as its own separate asset and thus, it was incorporated it into the physical environment branch whereby any object of interest is provided by the environment. Since the activity model was developed near the end of all of the explored techniques, it necessitated refinement of all of the other models. This is why a modeler need not wait until the end of a modeling effort to perform some simulations, as time could be saved if undesired system behaviors are identified early.

These insights also provide opportunities to simplify the models as necessary. The reality is that the entire process of modeling is iterative, so there is no guarantee that such insights will occur regardless of the order in which different models are constructed. Nevertheless, this particular insight was an important one because the refinement process that occurred as a result made every model easier to understand, more streamlined and readable, and simpler to trace for validity and continuity from technique to technique. There are many more insights yet to be gained by a diagram like Figure 17 and any action decompositions on which a modeler may wish to focus. For now, it stands as a solid DRM baseline activity simulation that can be easily adapted as desired for further understanding.

THIS PAGE INTENTIONALLY LEFT BLANK

VI. CONCLUSION

The best way to summarize this study is to go back to the objectives from Chapter II and see whether the modeling effort achieved the goals set forth.

The first objective was to find the modeling technique or language that best captures the complexities and realities of the SAR mission. A reader of this study will no doubt have noticed that during all of the modeling discussion in Chapter V, there was never an identification of a particular technique that was “better” than any of the others. The reason for this became apparent about halfway through the model development process because it turned out that the choice of modeling technique is wholly dependent on what kind of information the modeler seeks to acquire about the system. The reality of all model-based systems engineering is that it is a cumulative effort. If holistic understanding of a system is required, then multiple model constructions will be necessary using a number of different techniques because each method uncovers different strengths and insights. Like system design in general, choosing a modeling path will be full of tradeoffs. If a model is simple to build, understand, and communicate to stakeholders, chances are that it will lack some detail that might be desired by individuals deeply involved in the design. If a model’s strength is portraying interactions between major system assets and actors, it may lack the ability to show independent functions and behaviors where the actors perform something on their own. These are two tradeoff examples that were encountered in this study, but there were many more identified in Chapter V because tradeoffs are inherent to any modeling effort. This is why it is imperative to develop clear objectives and methodology at the onset of any modeling excursion. Without a robust plan, it is easy to get lost in the process because there are so many different techniques from which to choose.

Therefore, the answer to the first objective about finding the best technique is that it depends. If the modeling objectives are simple and the system is not complex, then perhaps a single technique will suffice. As systems become more complex, however, more effort is needed. Since a SAR architecture is an immense system of systems, even attempting to understand a single asset’s interactions, like the OSC, will require a great deal of understanding of all the other assets' behaviors in the system. Inevitably this will require multiple levels of modeling and decomposition in many techniques in order to accurately capture all the intricacies of the design. Limiting the modeling effort to only one technique runs the risk of missing out on key insights that are possible by translating a system model into another language or technique. This study proved the value of such iteration a number of times with the object of interest asset in particular, and those insights would not have been possible without the cumulative and iterative nature of the overall effort. Ultimately, the best technique will always involve developing clear objectives for the modeling and then having the requisite familiarity with the collection of available constructs so the development can satisfy the objectives.

The second objective of this study was to develop a robust and flexible base model for future implementation. This objective was important from the beginning, which is why a generic mission narrative and set of general rules in the DRM were the basis for the entire modeling effort. Keeping the base DRM generic forced the subsequent models to follow suit, meaning that now any specific scenario can potentially plug into one or all of the models in order to gain insight on the architecture's (or a specific asset's) performance given a set of mission parameters. This was the point of presenting a number of specific OPSITs in Chapter III. Using the data from any or all of the mission OPSITs would allow a modeler to assess the performance of a new procedure, an improved asset capability, different command structure, or any desired aspect of the architecture. Since the different modeling techniques each have something unique to offer, the flexibility is nearly unlimited for a modeler wishing to investigate important system interactions and even executable simulations.

The third objective involved ascertaining if there were any improvements to be made for the various modeling techniques in order to enhance their capabilities or make them easier to utilize. Many of the strengths and limitations of each technique explored in Chapter V were included in the individual model assessment sections and are summarized at the end of this chapter in Table 4. The discoveries that were made tie in largely with the findings for the first objective because oftentimes an identified limitation in a particular technique is covered by another technique. The positive aspects of having so many different models to choose from is that the models afford a cumulative effort that paints a picture of an entire system, depending on which aspects of the design a modeler wishes to focus. It is not desirable for one modeling technique to act as a "one-stop-shop" because the use of such a construct would be very complicated and the outputs would undoubtedly be difficult to understand for anyone outside of the area of expertise. Recalling what happened in the Figure 15 spider diagram where too many interactions and decompositions were portrayed, it is not hard to imagine the kinds of unreasonable outputs that would occur if a single modeling technique attempted to capture the entirety of important interactions in a complex system. This is not to say, of course, that there are not some individual improvements that could potentially occur with some of the techniques already in existence. Perhaps some extra simulation power would be beneficial for the action diagrams to further evaluate decision points, multiple recurring loops, or other potential scenario disruptions that more accurately mimic the uncertainties in the real world. Perhaps Monterey Phoenix requires a way to assign values to its executions to evaluate event traces on a deeper level. Suffice it to say that for this study's sampling of modeling techniques, most of the perceived limitations in any one model were covered by another. If a modeler cannot find the right fit for his or her objectives using available techniques in existence, then the modeler must explore changes to current constructs or create something customized to the intent. This is how modeling techniques evolve, once again proving that iteration is a key component to not only using model-based systems engineering for system evaluation but for the creation and redesign of the constructs themselves.

The fourth and final objective of this study was to gain insight and understanding into how model-based systems engineering pertains to real-world problems. Since this objective was already weaved into the first three objectives, many of the salient points have already been discussed. Nevertheless, it bears mentioning again that a great deal of the insight and understanding gained throughout this study has been applicable not only to potential engineers and designers, but to mission planners and operators as well. Each stakeholder group naturally will be interested in different facets of the model development, but there is no denying the flexibility of model-based systems engineering for a variety of uses well beyond the design lab.

In the grand scheme of model-based systems engineering, the development presented in this study is a small sampling of the techniques and constructs that exist for understanding and simulating complex system functions and behaviors. Tackling the SAR architecture, even on the generic level, is a massive undertaking due to the complexity and dynamic nature of the assets operating in the mission space. Nevertheless, model-based systems engineering has allowed for a successful breakdown of the architecture and mission space in order to identify key interactions and behaviors in the system. Recalling the capability need statement presented in Chapter I, civilian and defense agencies need an accurate and effective means of modeling SAR operational architectures across multiple scenarios in order to assess current and future capabilities so that persons in distress can receive aid in the shortest time possible. Utilizing the work presented in this study as a starting point, there are almost limitless possibilities of exploration regarding various aspects of the SAR architecture and how it might perform. These insights are crucial for the evolution of procedures, asset capabilities, and system infrastructure. The SAR architecture is no different than many other complex systems and the key to understanding critical interactions for today and the future must always include a robust model-based systems engineering effort.

Table 4. Model Summary of Features Table

TECHNIQUE	FEATURES
Sequence Diagram	• Simple graphical representation of more complex processes • Intuitive organization • Show interactions between assets • Interactions are known and sequential • Easy to translate into other models
IDEF0	• Detailed system activities for functional modeling • Concise models through abstraction • Not sequential • Multiple decompositions simplify complex high-level diagrams • Element consistency across multiple views aid in traceability and understanding
Hierarchy Diagram	• Simplify complex system by breaking it down • Used for functions or physical components • Flexibility for brainstorming • Solution-neutral on higher levels • Condensed views show entire system in one diagram
Monterey Phoenix	• Behavior modeling through multiple event traces • Powerful automatic generation of event traces using code • Exposes unexpected and undesired behaviors • Models easily refined by changing lines of code • Portray multiple scenarios with decision points • Produces simple and traceable graphical representations
Spider Diagram	• Extremely flexible brainstorming and planning tool • Can show physical and functional relationships on the same diagram • Easy to create, understand, and explain
Activity Model	• Executable system representations • Portray individual asset actions even if they have no interaction with other system assets • Simulations output real data values on system performance • Show how components interact over time • Show resource consumption • Ability to execute multiple decision points and activity loops • Useful for validating other MBSE models

THIS PAGE INTENTIONALLY LEFT BLANK

APPENDIX

This appendix shows traceability and connectivity for some selected models that were presented in Chapter V. The reason for a traceability assessment is to validate the continuity of the DRM mission narrative as it is translated from one model to another. The traceability assessment begins with the same sequence diagram from Chapter V. Each interaction is given a number from 1 to 21 inside of a red box so the sequence can be identified and traced through an asset diagram, the action diagram, and the high-level IDEF0. Of note, the asset diagram was not presented in Chapter V, but exists here to show high-level interactions between the major assets and the traceability connections for their relationships with each other.

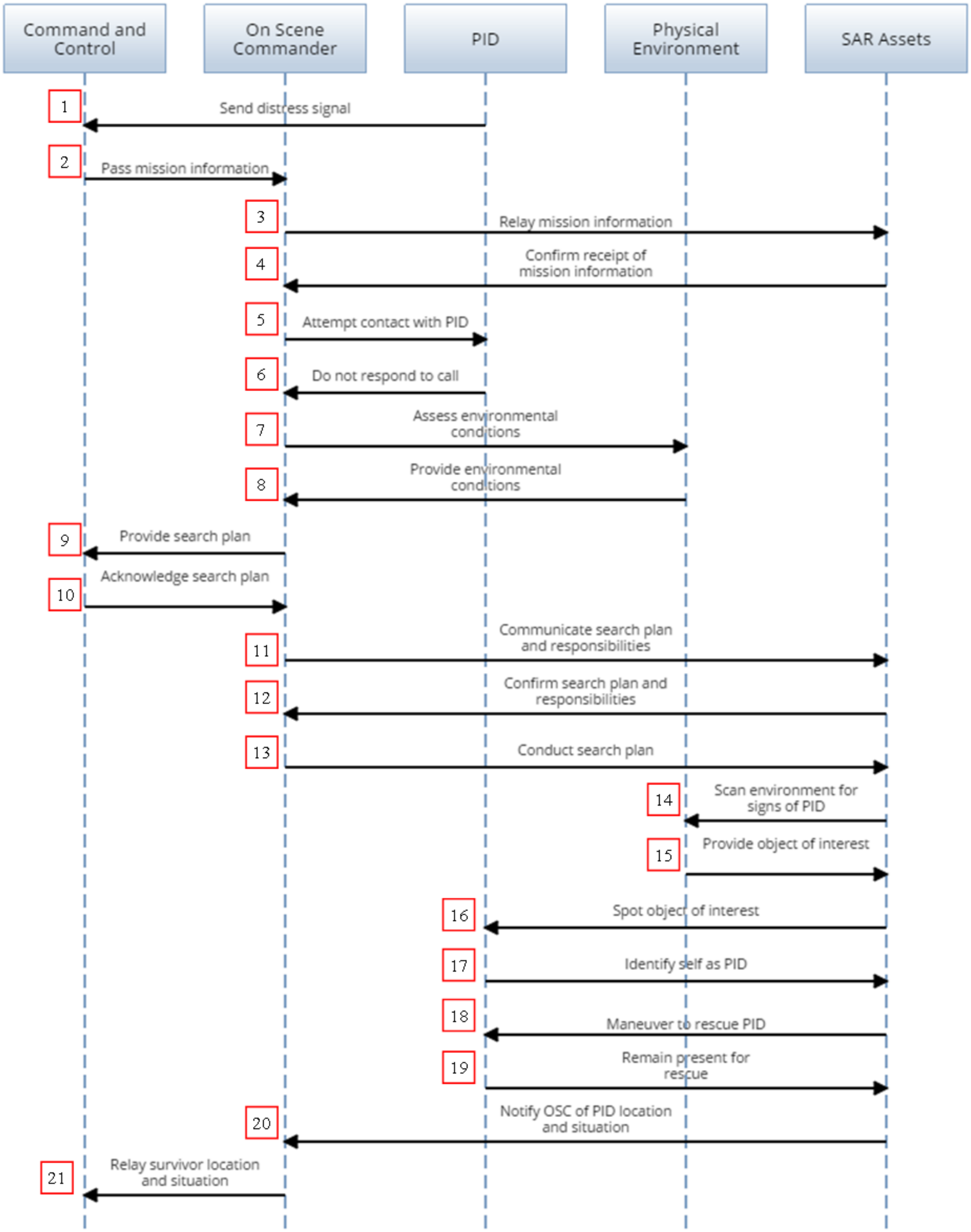

Figure 20. Appendix Sequence Diagram Traceability

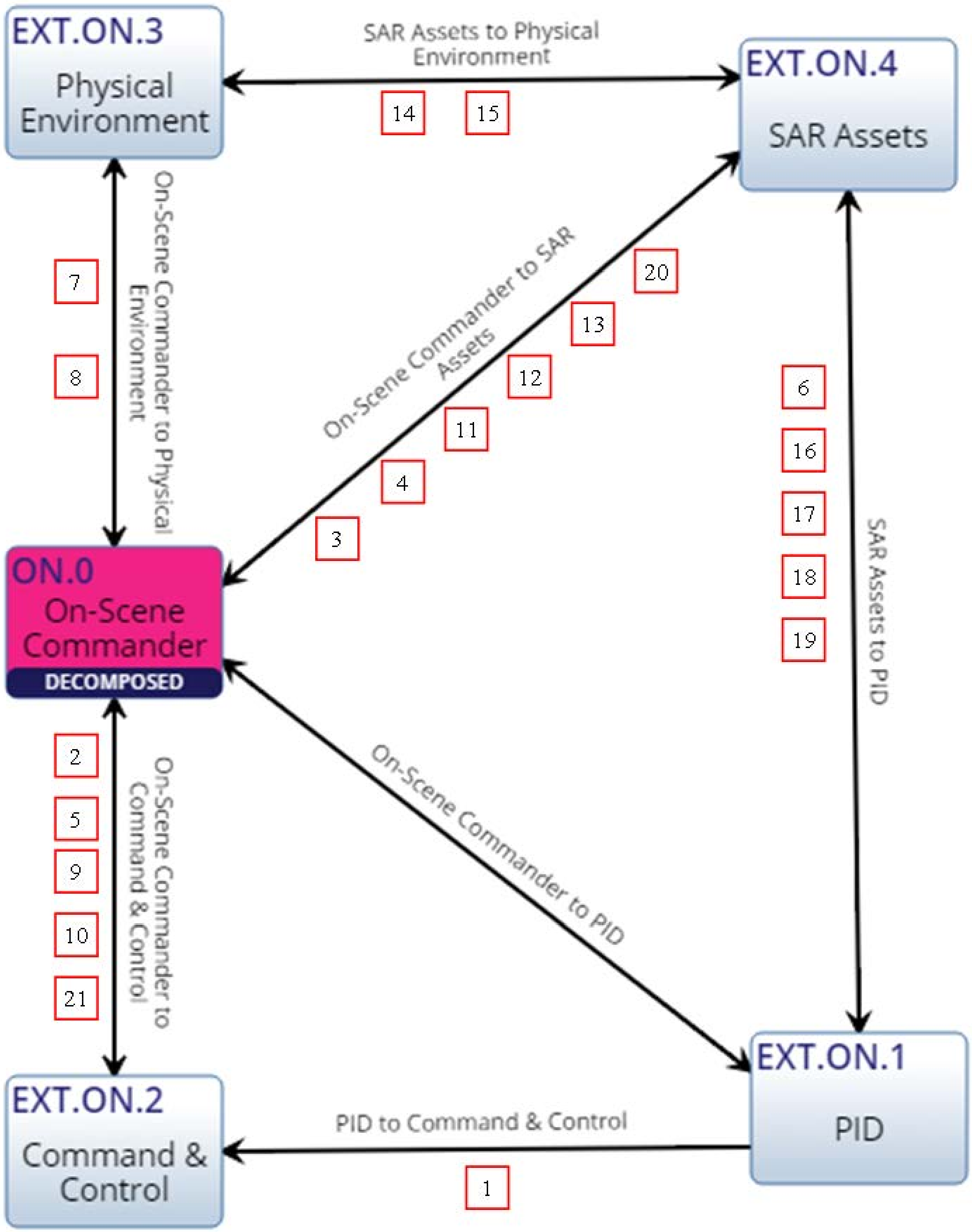

Figure 21. Appendix Asset Diagram Traceability

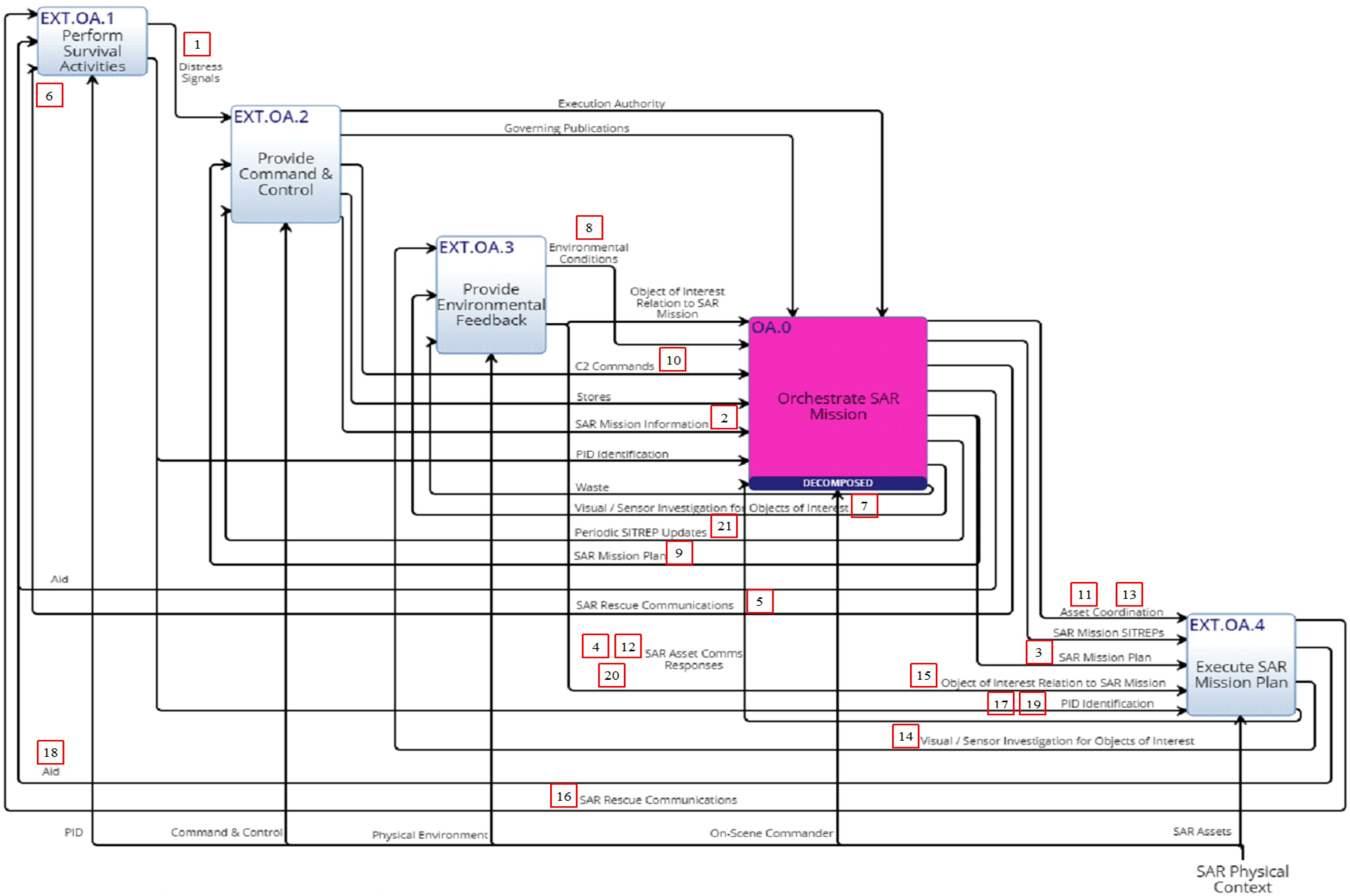

Figure 22. Appendix IDEF0 Diagram Traceability

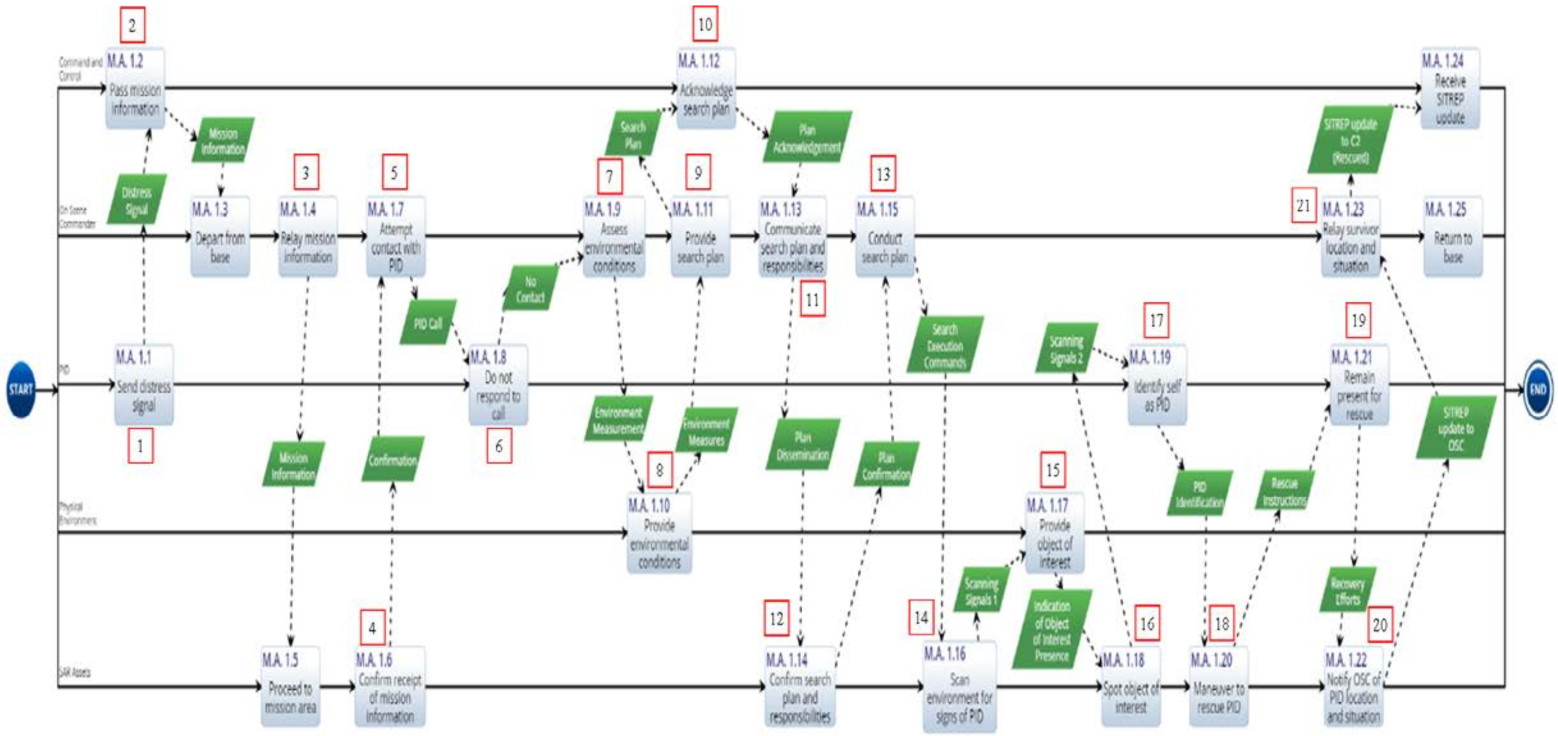

Figure 23. Appendix Action Diagram Traceability

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF REFERENCES

Auguston, Mikhail. 2015. "Monterey Phoenix: System and Software Architecture Modeling Language (Version 2.0)." Naval Postgraduate School, Department of Systems Engineering. Monterey, CA.

Bell, Donald. 2004. "UML Basics: The Sequence Diagram." IBM Corporation Rational Edge Articles. IBM developer Works. February 16. http://www.ibm.com/developerworks/rational/library/3101.html.

Department of Defense Systems Management College. 2001. *Systems Engineering Fundamentals*. Fort Belvoir, VA: Defense Acquisition University Press.

Farah-Stapleton, Monica, and Mikhail Auguston. 2013. "Behavior Modeling of Software Intensive System Architectures." *Procedia Computer Science* 20: 270–276. http://www.sciencedirect.com/science/article/pii/S1877050913010727.

Giammarco, Kristin, Spencer Hunt, and Clifford Whitcomb. 2015. "An Instructional Design Reference Mission for Search and Rescue Operations." Naval Postgraduate School, Department of Systems Engineering. Monterey, CA.

International Civil Aviation Organization and International Maritime Organization. 2013. *International Aeronautical and Maritime Search and Rescue Manual*. London: IMO.

International Maritime Organization, 2015. *IAMSAR Manual*. London: IMO. http://www.imo.org/en/OurWork/Safety/RadioCommunicationsAndSearchAndRescue/SearchAndRescue/Pages/IAMSARManual.aspx

Johnson, Paul, and Lee, Shelton. 2011. "DoDAF V2.0 In Action: Search and Rescue Example User Experience and Analysis." PowerPoint Slides presentment at Department of Defense Enterprise Architecture Conference, April 2011. http://www.slideshare.net/pjohnson28/do-d-ea-conference-sar-dodaf-in-action-apr13apwj.

Knowledge Based Systems Incorporated. 2010. *IDEF Family of Methods: A Structured Approach to Enterprise Modeling & Analysis*. Station, TX: Knowledge Based Systems. http://www.idef/IDEF0.htm.

Major, R.H. 1859. *Early Voyages to Terra Australis, Now Called Australia*. London: Hakluyt Society.

Metron Incorporated. 2007. "Simulator and Search Planner for the USCG SAROPS: Search and Rescue Optimal Planning System." Reston, VA: Metron. Accessed January 8, 2015. http://www.sarapp.com/docs/SAROPS%20Description.pdf.

United Nations (UN) General Assembly. 1979. *International Convention on Maritime Search and Rescue 1979*. Official Records, Vol. 1405, 1–23489.

United States National Search and Rescue Committee (NSARC). 2007. *National Search and Rescue Plan of the United States.* Official Records. Washington, DC: NSARC.

United States National Search and Rescue Committee (NSARC). 2000. *United States National Search and Rescue Supplement to the International Aeronautical and Maritime Search and Rescue Manual*. Washington, DC: NSARC.

Visual Paradigm International. 2015. "UML Modeling: Activity Diagram." Accessed July 21, 2015. http://www.visual-paradigm.com/VPGallery/diagrams/Activity.html.

INITIAL DISTRIBUTION LIST

1. Defense Technical Information Center
   Ft. Belvoir, Virginia

2. Dudley Knox Library
   Naval Postgraduate School
   Monterey, California